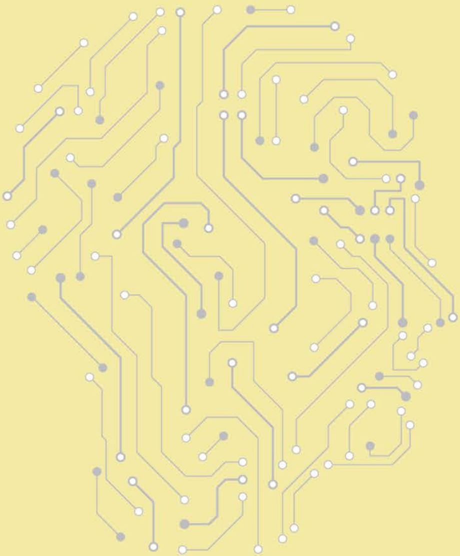
图灵原创

人工智能简史

A Brief History of Artificial Intelligence

尼克 著



中国工信出版集团 人民邮电出版社 POSTS & TELECOM PRESS

数字版权声明

图灵社区的电子书没有采用专有客户端，您可以在任意设备上，用自己喜欢的浏览器和PDF阅读器进行阅读。

但您购买的电子书仅供您个人使用，未经授权，不得进行传播。

我们愿意相信读者具有这样的良知和觉悟，与我们共同保护知识产权。

如果购买者有侵权行为，我们可能对该用户实施包括但不限于关闭该帐号等维权措施，并可能追究法律责任。

尼克，早年曾任职哈佛和惠普；后创业投资，往返于大陆和硅谷。无论忙闲不忘读书写字，作品多发表于《上海书评》，并有著作《UNIX系统V内核剖析》和《哲学评书》。

TURING 图灵原创

人工智能简史

A Brief History of Artificial Intelligence

尼克 著

人民邮电出版社
北京

图书在版编目（C I P）数据

人工智能简史 / 尼克著. -- 北京 : 人民邮电出版社，2017.12（2018.3重印）
（图灵原创）
ISBN 978-7-115-47160-4

Ⅰ. ①人… Ⅱ. ①尼… Ⅲ. ①人工智能－简史 Ⅳ. ①TP18

中国版本图书馆CIP数据核字(2017)第263722号

内 容 提 要

本书全面讲述人工智能的发展史，几乎覆盖人工智能学科的所有领域，包括人工智能的起源、自动定理证明、专家系统、神经网络、自然语言处理、遗传算法、深度学习、强化学习、超级智能、哲学问题和未来趋势等，以宏阔的视野和生动的语言，对人工智能进行了全面回顾和深度点评。

本书作者和书中诸多人物或为师友或相熟相知，除了详实的考证还有有趣的轶事。本书既适合专业人士了解人工智能鲜为人知的历史，也适合对人工智能感兴趣的大众读者作为入门的向导。

◆ 著　　　　尼　克
责任编辑　张　霞
责任印制　彭志环

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路11号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京　　　　　　印刷

◆ 开本：880×1230　1/32
印张：10.125
字数：243千字　　　　2017年 12 月第 1 版
印数：22 001 – 27 000册　　　　2018年 3 月北京第 4 次印刷

定价：49.00元

读者服务热线：(010)51095186转600　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

前　言

历史素有两种写法：以人为主和以事为主。所有的传记都是以人为主的；而各种专史，如战争史，则多以事为主。所谓历史是人民创造的还是英雄创造的，我个人的偏好还是以人为本。八卦的历史，读者自然喜欢，对作者也有好处，就像一战后英国首相劳合·乔治对他的耶路撒冷总督说的那样：有争执，咱们政治家才派得上用场，如果他们停下来不打了，你就失业了。

人工智能到底是什么？给一门学科界定范围很难，尤其是这门学科还在快速变化中。即使是数学这样的成熟学科，有时我们也理不清边界，而像人工智能这样朝令夕改的，更是不容易闹清楚了。人工智能的定义素无共识。在大学里，机械系、电子系、计算机系，甚至哲学系都有人干人工智能。让这些人对这门学科取得共识谈何容易。从实用主义（哲学的“实用主义”，不是日常用语“实用主义”）看，一个学科就是学科共同体共同关注的东西。有些毛边可以宽容，演变。这种外延式的定义要比从上帝视角给一个内涵式定义更为实用。

一般认为，人工智能起源于 1956 年在达特茅斯学院召开的夏季研讨会。国内关于达特茅斯会议和神经网络早期历史的各种段子很多源于我几年前的两篇博客，后来被《上海书评》转发。经过修订，我把它们重新编

为本书的两章："达特茅斯会议：人工智能的缘起"和"神经网络简史"。"计算机下棋"一章的大部分也在《南方周末》发表过。"自动定理证明兴衰纪"的核心内容在《中国计算机学会通讯》连载过。

明尼苏达大学的查尔斯・巴贝奇研究所一直在做计算机科学的口述历史，采访了很多对计算机科学有影响的人，其中有相当一批是人工智能学者。大部分的采访都有录音。除了翻阅各种文献外，我听了近 100 小时的采访录音，许多人工智能老一代革命家临终时话都说不利索，听这种东西除了兴趣，还得有体力。

图灵大概是第一个对智能做出深刻思考的智者。他 1936 年的文章"可计算的数"奠定了计算机科学的理论和实践基础，也把相关的哲学思考推进了一大步，以至于哲学家蒙克（Ray Monk）把他列为有史以来最伟大的十位哲学家之一。图灵 1950 年在哲学杂志《心》（*Mind*）上发表的文章"计算机与智能"是传世之作，但这篇文章没有靠谱的中文翻译，我将我的译文和一篇图灵小传附在书后作为附录。图灵小传的一个早期版本曾出现在我的《哲学评书》一书中，但新版本融入了一些近几年关于图灵研究的新成果。"人工智能"这个词组的出现和达特茅斯会议有关。但英国学术圈在 1956 之前和之后的很长一段时间一直在用"机器智能"的说法，这和图灵 1950 年的文章有关。一般认为，这篇文章是这个学科的源头。但后来发现图灵 1948 年在英国国家物理实验室（NPL）写过一个内部报告，题为"智能机器"，其中提到了"肉体智能"（embodied intelligence）和"无肉体智能"（disembodied intelligence）的区分。机器人学家布鲁克斯（Rodney Brooks）认为图灵 1948 年的报告比图灵 1950 年的文章更加重要，它从某种意义上预示了后来符号派和统计派之争。这段历史我也列在附录里，放在图灵小传之后，因为我觉得先读读图灵的生平也许会有助于理解他的思想。

本书每一章几乎都可单独阅读，大部分内容，对于受过高中教育的人，应该都不难懂。但第 10 章是个例外，这一章企图以严肃的态度探讨人工智能。我以一种浓缩的方式讲述了图灵机、丘奇–图灵论题、相似性原则和超计算。没有计算理论，很多人工智能的基础问题实在是拎不清。如果读者觉得吃力，可以跳过这一章。

我常用的一种历史研究工具是谷歌的 Ngram。谷歌扫描了三千多万本书，把书中出现的词组的词频统计结果公布。以时间为横轴、词频为纵轴画一条曲线，就可看出特定的词在不同历史时间段的兴衰，从而得出某些洞察。例如，通过比较“United States are”和“United States is”在历史上出现的频率，就可看出美国人是何时开始认同美国作为一个统一的国家的。很明显，南北战争之后，“United States is”开始变得更常用。我们通过统计若干人工智能中关键词的 Ngram，可以感知人工智能在不同阶段的宏观发展。我曾经写过一篇“计算历史学”（见《哲学评书》）介绍 Ngram。大数据为历史学提供了有力而令人信服的工具。

科普有一种写法：用一些貌似通俗的语言去解释复杂的原理。我一直不大相信这种方法，无论作者是内行还是专业科普作家。我压根就没见过一本可以把量子力学解释清楚的科普书。即使简单如图灵机，也鲜有适当的普及读物。倒是那些讲历史和八卦的书引人入胜，安德鲁・霍奇斯的《艾伦・图灵传：如谜的解谜者》是内行写作的典范，而数学家所罗门・费佛曼的太太安妮塔・费佛曼的两本逻辑学家传记是我心目中的标杆。戴森（Freeman Dyson）一直是我喜欢的作者，他也时不时为《纽约书评》写写八卦，趣味和我接近，我总是从阅读他的文章的过程中收获良多。即使我不懂他的数学和物理的领域，也能时有洞察。我的书单上还有蒙克的所有传记，它们既高级又有趣。就像蒙克所说，历史可以帮助内行

了解知识的进化并获得新的视角，同时也为外行人了解专业知识提供入门的台阶或向导。

读大科学家写的科普著作，最有意思的倒不是那些对成熟思想的通俗叙述，而是那些对不成熟看法的披露，还有不好意思写到正经学术论文里的自负和牢骚。恰因为这个原因，我也喜欢多依奇（David Deutsch）的几本书。

我们很少有机会在学科发展之初就能把学科脉络梳理清楚。过去有过几个这样的时间段，例如 1900 年到 1950 年的逻辑学，1945 年到 2000 年的分子生物学和 1950 年到当下的语言学。本书除了想梳理始于 20 世纪 40 年代的人工智能的历史外，还有一个作者隐含的心愿：作为人工智能的科普。哈代曾说科学（尤其是数学和理论物理，也许还有理论计算机科学）和艺术的原创需要一等的智力，解释和欣赏（例如乐评家和书评家）是二等的智力活儿。本书假想的对象是那些有能力但又是外行的人。丘成桐曾说（大意）：要想做大学问，必须先培养对学问的感情。除了科普，我还希望能帮助一小撮内行人或准内行人培养感情。我尽可能地列出了相关的参考文献供进一步学习。人工智能毕竟不是超弦理论，凭着一些智力还是可以自学的。

本书写作得到白硕、陈利人、宫力、洪涛、李维、刘江、马少平、毛德操、施水才、宣晓华和赵伟等诸位师友的帮助和指点，特此致谢。乌镇智库的同仁为本书提供了必要的数据，我的助理冰冰为我提供了多方面的支持，一并谢过。

目　录

第 1 章
达特茅斯会议：人工智能的缘起

What is past is prologue.

过去只是序幕。

——William Shakespeare（莎士比亚）

1. 背景

现在一说起人工智能的起源，公认是 1956 年的达特茅斯会议。殊不知还有个前戏，1955 年，美国西部计算机联合大会（Western Joint Computer Conference）在洛杉矶召开，会中还套了个小会：学习机讨论会（Session on Learning Machine）。讨论会的参加者中有两个人参加了第二年的达特茅斯会议，他们是塞弗里奇（Oliver Selfridge）和纽厄尔（Allen Newell）。塞弗里奇发表了一篇模式识别的文章，而纽厄尔则探讨了计算机下棋，他们分别代表两派观点。讨论会的主持人是神经网络的鼻祖之一皮茨（Walter Pitts），他最后总结时说："（一派人）企图模拟神经系统，而纽厄尔则企图模拟心智（mind）……但殊途同归。"这预示了人工智能随后几十年关于"结构与功能"两个阶级、两条路线的斗争。

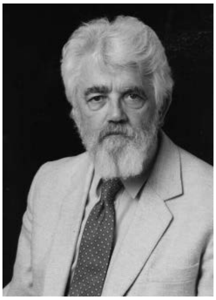
开聊达特茅斯会议之前，先说说 6 个最关键的人。首先，会议的召集者麦卡锡（John McCarthy）当时是达特茅斯学院的数学系助理教授。

1954 年，达特茅斯学院数学系同时有 4 位教授退休，这对达特茅斯这样的小学校而言真是不可承受之轻。刚上任的年轻系主任克门尼（John Kemeny）之前两年才在普林斯顿大学逻辑学家丘奇（Alonzo Church）门下取得了逻辑学博士，于是跑到母校求援。这么说起来，克门尼算是图灵的师弟，他战时和物理学家费曼一起工作，还一度当过爱因斯坦的数学助理，后来一头扎在计算机研究里，和麦卡锡一起琢磨出了分时系统。他 1955 年在《科学美国人》杂志上写过一篇文章“把人看作机器”（Man Viewed as a Machine），介绍了图灵机和冯诺伊曼[①]的细胞自动机（最早叫“自生机”），文章的简介提到“肌肉机器”（muscle machine）和“大脑机器”（brain machine）。所谓“大脑机器”就是人工智能的另一种说法而已。克门尼最为人知的工作应该是发明了老少咸宜的编程语言 BASIC。现在估计已经没人知道 BASIC 语言发明人曾是 LISP 语言发明人的老板。克门尼是天生的官僚，后来位居达特茅斯学院数学系主任和校长，美国三里岛核电站出事时，总统委托他当调查委员会主席，这是后话。

克门尼从母校数学系带回了刚毕业的 4 位博士前往达特茅斯学院任教，麦卡锡是其中之一。麦卡锡后来发明的 LISP 语言中最重要的功能 Eval 实际就是丘奇的 λ 演算，而且他后半生致力于用数理逻辑把常识形式化。大家由此猜测他可能也是丘奇的学生，但其实不是，他学的压根就不是逻辑。他的老师是失去双手的代数拓扑学家莱夫谢茨（Lefschetz）。但麦卡锡对逻辑和计算理论一直有强烈兴趣，他 1948 年本科毕业于加州理工学院，在学校主办的 Hixon 会议上听到冯诺伊曼关于细胞自动机的讲座，后来他刚到普林斯顿大学读研究生时就结识了冯诺伊曼，在老冯影响下开始对在计算机上模拟智能产生兴趣。

---

① 我故意没有在“冯”和“诺伊曼”之间加那个讨厌的点儿，因为在更多时候，查找参考文献时，他的姓是列在 V 下，而不是 N 下。



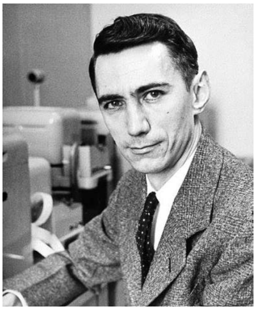
麦卡锡（1927—2011）

达特茅斯会议的另一位积极的参加者是明斯基。他也是普林斯顿大学的数学博士，和麦卡锡在读书时就相熟。他的主业也不是逻辑，尽管他后来写过一本很有影响力的计算理论的书，还培养过好几个计算理论的博士，其中就有图灵奖获得者布鲁姆（Manual Blum）。布鲁姆目前和他老婆（Lenor Blum，就是实数计算模型 BSS 的 B）、儿子一家三口都在卡内基梅隆大学任教。明斯基的理论情结和丘奇关系也不大，他的老师塔克（Albert Tucker）是莱夫谢茨的学生，主要做非线性规划和博弈论，多年来担任普林斯顿大学数学系主任，出身数学世家，儿子、孙子也都是数学家。按辈分论，麦卡锡还是明斯基的师叔。塔克的另一名出色的学生后来得了诺贝尔经济学奖，他就是心灵美丽的纳什。纳什比明斯基小一岁，但比他早 4 年拿到博士学位，也算是明斯基的师兄了。明斯基的博士论文是关于神经网络的，他在麻省理工学院 150 周年纪念会议上回忆说是冯诺伊曼和麦卡洛克（Warren McCulloch）启发他做了神经网络。有人还找过他麻烦，质疑说神经网络的研究算数学吗，倒是老冯力挺说：现在不算，但很快就得算。倒是明斯基自己后来和神经网络结下梁子，那段故事见本书第 5 章“神经网络简史”。明斯基的熟人都认为他是无所不通的天才，他的忘年交沃尔弗拉姆（Stephen Wolfram）称，他晚年计划写本神学的书，但去世时书还没影子。

塞弗里奇被后人提及不多，但他真是人工智能学科的先驱，他在麻省理工学院时一直和神经网络的开创人之一麦卡洛克一起在维纳（Norbert Wiener）手下工作，他是维纳最喜欢的学生，但没读完博士学位。维纳《控制论》一书的第一个读者就是塞弗里奇。塞弗里奇是模式识别的奠基人，他写了第一个可工作的 AI 程序。他后来在麻省理工学院参与领导 MAC 项目，这个项目后来一分为二：计算机科学实验室和人工智能实验室。但分久必合，现在这两个项目又合并了，变成了 MIT CSAIL。顺便给女读者添点料：塞弗里奇的爷爷就是英国第二大百货店塞尔福里奇（Selfridges）的创始人。所谓“顾客永远是对的”（The customer is always right.）就出自塞尔福里奇，他本是美国人，后到英国创业，发财后老婆就死了，于是勾搭上一对匈牙利双胞胎歌舞演员，出入赌场，赔光了家业。他的故事 2013 年还被有意思的英国人拍成了电视剧。塞尔福里奇百货几经周转，现在的主人是美国百货公司希尔斯（Sears）。塞尔福里奇百货和隔壁的哈罗德百货支撑着牛津街的零售业，现在大概一半顾客来自中国。

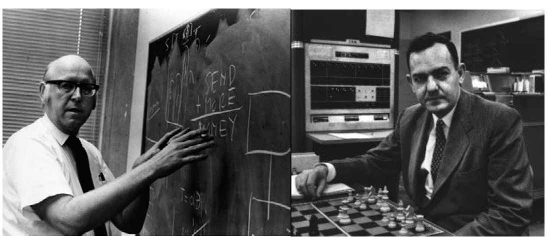
信息论的创始人香农（Claude Shannon）被麦卡锡拉大旗做虎皮也请到会上打酱油。其实麦卡锡和香农的观点并不一致，平日相处也不睦。香农的硕士、博士论文都是讲怎么实现布尔代数的，当时麻省理工学院校长布什（Bush）亲自指导。博士毕业后他去了普林斯顿高等研究院，曾和数学家外尔（Hermann Weyl）、爱因斯坦、哥德尔等共事。战争中，他一直在贝尔实验室做密码学的工作，图灵在 1943 年曾秘访美国，和同行交流破解德国密码的经验，其间和香农曾有会晤，一起聊过通用图灵机。战后香农去英国还回访过图灵，一起讨论过计算机下棋。香农内向，从没说过这段往事，直到 1982 年接受一次采访时才提起。1950 年香农在《哲学杂志》发表过一篇讲计算机下棋的文章，为计算机下棋奠定了理论基础。香

农比其他几位年长十岁左右，当时已是贝尔实验室的大佬。



香农（1916—2001）

另外两位重量级参与者是纽厄尔和司马贺（Herbert Simon）。纽厄尔是麦卡锡和明斯基的同龄人，他硕士也是在普林斯顿大学数学系读的，按说普林斯顿大学数学系很小，他们应有机会碰面，但那时纽厄尔和他俩还真不认识。他们的第一次见面，纽厄尔回忆是在 IBM，而麦卡锡回忆是在兰德公司。纽厄尔的硕士导师就是冯诺伊曼的合作者、博弈论先驱摩根斯顿，纽厄尔硕士毕业后就迁往西部加入著名智库兰德公司。他在兰德开会时认识了塞弗里奇，并受到对方做的神经网络和模式识别的工作的启发，但方法论走的却完全是另一条路。



纽厄尔（1927—1992）与司马贺（1916—2001）

司马贺比他们仨都大 11 岁（怀特海比罗素也大 11 岁），那时是卡内基理工学院（卡内基梅隆大学的前身）工业管理系的年轻系主任，他在兰德公司学术休假时认识了纽厄尔。司马贺后来把纽厄尔力邀到卡内基梅隆

大学，并给纽厄尔发了个博士学位，开始了他们终生的合作。

纽厄尔和司马贺的合作是平等的，司马贺是纽厄尔的老师，但他们合作的文章署名都是按字母顺序纽在前司马在后，每次他们受邀去演讲，都是轮流。司马贺每次见到别人把他名字放到纽厄尔之前时都纠正。他们共享了 1975 年的图灵奖，三年后司马贺再得诺贝尔经济学奖。纽厄尔和司马贺代表了人工智能的另一条路线：符号派。他们后来把他们的哲学思路命名为“物理符号系统假说”。简单地说就是：智能是对符号的操作，最原始的符号对应于物理客体。这个思路和英美的经验主义哲学传统接近。他们和当时的数学系主任、第一届图灵奖获得者珀里思（Alan Perlis）一起创立了卡内基梅隆大学的计算机系，从此，卡内基梅隆大学成为计算机学科的重镇。

2. 达特茅斯会议



会议原址：达特茅斯楼

1953 年夏天，麦卡锡和明斯基都在贝尔实验室为香农打工。香农那时的兴趣是图灵机以及是否可用图灵机作为智能活动的理论基础。麦卡锡向香农建议编一本文集，请当时做智能研究的各位大佬贡献文章，这本文集直到 1956 年才以《自动机研究》（*Automata Studies*）为名出版，这个书名最后是香农起的，他不想花里胡哨，但麦卡锡认为这没有反映他们的初衷。

文集的作者有两类人，一类是逻辑学家（后来都变成计算理论家了），如丘奇的两位杰出学生戴维斯和克里尼，后者的名著《元数学导论》在国内有逻辑学家莫绍揆先生的译本。明斯基、麦卡锡也都有论文录入，香农本人贡献了一篇讲只有两个内部状态的通用图灵机的文章，文集录入的一篇冯诺伊曼的论文后来开创了容错计算。文集的另一类作者几乎都是维纳的信徒，如阿什比（Ross Ashby）等，以控制论为基础。麦卡锡素不喜控制论和维纳，既不想把维纳当老大，也不愿和他见面争执，其中原因不详，或许和维纳与麦卡洛克吵翻了有关。麦卡洛克和皮茨这两位为维纳《控制论》思想贡献多多的人物，在维纳的自传里压根没被提及。麦卡锡同时又觉得香农太理论，当时他想自立门户，只对用计算机实现智能感兴趣，于是他筹划再搞一次活动。从香农后来接受的采访来看，他对维纳也没有多少尊重，他觉得自己创立的信息论和维纳一点关系也没有。但维纳却认为香农受到他的影响，香农认为维纳的这种错觉来源于维纳根本不了解信息论。

1955 年夏天，麦卡锡到 IBM 打工（美国教授都是 9 个月工资，如果没有研究经费，夏天要自己觅食），他的老板是罗切斯特（Nathaniel Rochester），罗切斯特是 IBM 第一代通用机 701 的主设计师，对神经网络素有兴趣。他们两人倒是挺对脾气，决定第二年夏天在达特茅斯搞一次活

动，遂说动了香农和当时在哈佛做初级研究员（Junior Fellow[①]）的明斯基一起给洛克菲勒基金会写了个项目建议书，希望得到资助。美国富豪还是有文化传统的，至少知道要资助好东西，值得中国土豪的后代学习。

麦卡锡给这个第二年的活动起了个当时看来别出心裁的名字：人工智能夏季研讨会（Summer Research Project on Artificial Intelligence）。普遍的误解是“人工智能”这个词是麦卡锡想出来的，其实不是。麦老晚年回忆也承认这个词最早是从别人那里听来的，但记不清是谁了。后来英国数学家伍德华（Philip Woodward）给《新科学家》杂志写信说他是 AI 一词的原创者，麦卡锡最早是听他说的，因为他 1956 年曾去麻省理工学院访问，见过麦卡锡并交流过。但麦卡锡的建议书 1955 年就开始用“人工智能”了，人老了回忆真不靠谱。当事人都已仙逝，这事恐怕要成悬案了。其实英国人最早的说法是“机器智能”（Machine Intelligence），这大概和图灵那篇“计算机与智能”有关。

大家对“人工智能”这个词一开始并没取得完全共识。很多人认为啥事一加“人工”就变味了。纽厄尔和司马贺一直主张用“复杂信息处理”这个词，以至他们发明的语言就叫 IPL（Information Processing Language）。他们从某种意义上说偏功能学派，也就是说找到智能的功能不一定非得依靠结构相同或相似。图灵机和递归函数等价，但结构完全不同，所以他们强调“信息处理”。他们俩一开始颇不喜“人工智能”几个字。1958 年，在英国国家物理试验室（NPL）召开了“思维过程机器化”（Mechanization of Thought Process）会议，达特茅斯会议的与会者麦卡锡、明斯基、塞弗里奇都参加了，此外还有致力于神经网络研究的麦卡洛克，以及英国的控

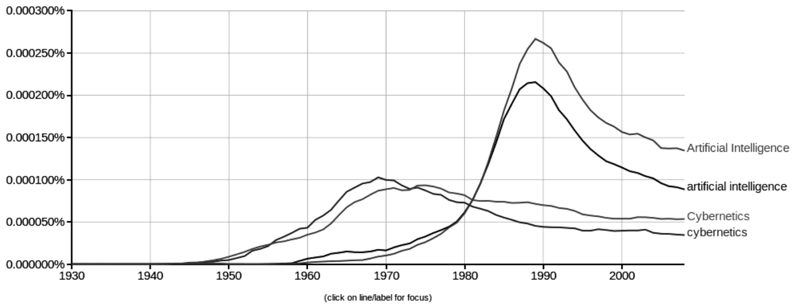
① 哈佛的 Fellow 还是挺值钱的，历史上人数不多，蒯因、王浩、库恩在变成正式教授之前都做过。乔姆斯基几乎在同时也是哈佛的 Fellow。

制论代表人物阿什比。两位编程语言的先驱也出席了：巴克斯（John Warner Backus）发表了一篇关于他新发明的语言 Fortran 的论文，但他后来一直是函数式语言的倡导者；美国海军女少将哈泊（Grace Hopper）的文章是讲第一个编译器的，这项工作导致了 COBOL 语言的诞生。中国也有女少将，也是码农。他俩论文的题目里都有 Automatic Programming 的说法，这在当时就是指高级语言编程，不能和后来人工智能中的自动编程搞混了。这次会上有人再提“人工思维”（Artificial Thinking）的说法。司马贺等人由此也逐渐接受了 AI 的说法，他晚年还写了本书《人工的科学》，倒是把 Artificial 这个词更加放大了。

3. AI 历史的方法论

历史研究方法有基于事件的和基于课题（issue）的。纽厄尔在 1981 年为一本颇为有料的文集《信息研究》贡献的一篇文章“AI 历史的智力课题”走了第二条路线。他的方法也挺有意思。他把 AI 历史当作斗争史，把历史分为两个阶级、两条路线的斗争，于是历史成了一串儿对立的议题，如模拟与数字，串行与并行，取代与增强，语法与语义，机械论与目的论，生物学与活力论，工程与科学，符号与连续，逻辑与心理等，在每一议题下有进一步可分的子议题，如在逻辑与心理下又有定理证明与问题求解等。

被提到最多的是人工智能与控制论。在 Google Ngram 里试试 Cybernetics 和 Artificial Intelligence 两个词在 Google Books 里出现的词频，可以看出学科的跌宕起伏。



"人工智能"与"控制论"词频对比

美国最早办的一批计算机相关的系科都创办于 20 世纪 60 年代中期，那时有些系直接叫"计算机科学系"，而有些则叫"计算机与信息科学系"，带"信息"的都有些"控制论"的背景，如麻省大学计算机与信息系的创办人就有维纳的学生阿比卜（Michael Arbib）。而密歇根大学则叫计算机与通讯科学系。这些系后来都改名叫计算机系了。而原来的图书馆系现在都纷纷改名叫信息科学系，如加州大学伯克利分校和华盛顿大学的图书馆学院都改名叫信息学院（School of Information），连"科学"都省了。但现在计算机系又有加载信息的趋势，麻省大学和加州大学尔湾分校近年又改名叫信息与计算机科学学院了。大概和现在深度学习及神经网络又峰回路转有关吧。倒是中国的学科简单，一直都有计算机和自动化之分，老死不相往来罢了。

"人工智能"这个词真正被共同体广泛认可是在十年后的 1965 年，在加州大学伯克利分校的欧陆派哲学家德雷弗斯（Hubert Dreyfus）发表了"炼金术与人工智能"一文之后。这篇文章一开始只是针对纽厄尔和司马贺的工作，几年后这篇文章演变成了那本著名的（或者被 AI 圈子称为"臭名昭著"的）《计算机不能干什么》一书，则是把整个 AI 当作靶子。欧陆派哲学家被人诟病数学和科学不通，但德雷弗斯有个数学家的兄弟，和他同

一年在哈佛得了应用数学博士，后来又同在加州大学伯克利分校教书，是动态规划的大家，还带过神经网络的博士。哥俩一个立场。有时一个共同体的形成并不是靠内部的团结，而是靠外部的反对。有意思的是，“炼金术与人工智能”一文是德雷弗斯在兰德公司工作时写就的。司马贺后来撰文猛批德雷弗斯，说他滥用兰德公司的标签。德雷弗斯后来抱怨他在麻省理工学院和哈佛食堂吃饭，所有做 AI 的人都躲他远远的。学术争执哪儿都一样。

麦卡锡和明斯基的建议书里罗列了他们计划研究的 7 个领域：(1) 自动计算机，所谓“自动”指的是可编程；(2) 编程语言；(3) 神经网络；(4) 计算规模的理论（theory of size of a calculation），这说的是计算复杂性，明斯基后来一直认为计算理论是人工智能的一部分，他早期对理论问题时不时会动动手，后来一手组建了麻省理工学院的计算理论队伍；(5) 自我改进，这个是说机器学习；(6) 抽象；(7) 随机性和创见性。

麦卡锡的原始预算是一万三千五百美元，但洛克菲勒基金会只批了七千五百美元。麦卡锡预计会有 6 位学界的人出席，会议应该支付每人两个月的薪水一千两百美元，由此可推算出麦卡锡、明斯基当时的年薪在八千美元左右，考虑通货膨胀和购买力，大概相当于 2016 年的七万多美元，真不算多，现在随便一个美国大学计算机系的教授薪水都远不止这个数。这个学科真是今非昔比啊。作为对比，司马贺 1949 年去卡内基梅隆大学的前身卡内基理工学院担任新成立的工业管理系系主任时的年薪是一万美元。

除了那六君子外，另外还有 4 人也参加了达特茅斯会议。他们是来自 IBM 的塞缪尔（Arthur Samuel）和伯恩斯坦，他们一个研究跳棋，一个研究象棋。达特茅斯的教授摩尔（Trenchard More）也参与了，他后来在工

业界混的时间长，少为外人所知。达特茅斯会议中一位被后人忽视的“先知”是所罗门诺夫（Solomonoff）。

和其他来来往往的人不同，所罗门诺夫在达特茅斯严肃地待了整整一个暑假。他 1951 年在芝加哥大学跟随费米得了物理硕士就到了麻省理工学院。但在芝加哥对他影响最大的是哲学家卡尔纳普（Paul Carnap）。有意思的是，神经网络的奠基者之一皮茨也受惠于卡尔纳普。司马贺的回忆录里也讲到自己在芝加哥时听卡尔纳普的课开始启蒙逻辑，从而开始对智能相关的问题感兴趣，但后来由于和定理证明逻辑派之间的冲突，司马贺就说自己的方法是在批判过度数学化和形式化。这么说来，人工智能的两大派——逻辑和神经网络——都发源于老卡。卡尔纳普那时的兴趣是归纳推理，这成为所罗门诺夫毕生的研究方向。所罗门诺夫后来结识了明斯基和麦卡锡，在他们的影响下研究逻辑和图灵机。达特茅斯会议时，他受麦卡锡“反向图灵机”和乔姆斯基文法的启发，发明了“归纳推理机”。他的工作后来被万能的苏联数学家柯尔莫格罗夫（Kolmogorov）独立地发明了一遍，就是现在俗称“柯尔莫格罗夫复杂性”和“算法信息论”的东西。中国的计算理论学者李明现在是这个领域的大牛，曾有专著。柯尔莫格罗夫 1968 年开始引用所罗门诺夫的文章，使得后者在苏联的名声比在西方更加响亮。所罗门诺夫的另一个观点“无限点”（Infinity Point）后来被未来学家库兹韦尔（Ray Kurzweil）改名“奇点”窃为己有。目前 AI 中广泛用到的贝叶斯推理也有着所罗门诺夫的开创性痕迹。他一生并没有大富大贵，大部分时间都是在自己的咨询公司 Oxbridge（牛津+剑桥，相当于汉语俗称“清北”）拿政府（空军、海军、ARPA 和 NIH——NIH 资助了很多 AI 研究）的研究经费，那公司只有他自己一个雇员。伦敦大学皇家哈洛威学院（Royal Holloway）后来在苏联学者领导下搞柯尔莫格罗夫奖，他是第一届获奖人，

并在那里兼职教授。他的学术自传 1997 年发表在计算理论杂志《计算机与系统科学》上。明斯基所谓 AI 孵化出计算理论的说法不无道理。

按照麦卡锡和明斯基的说法，这十个人参加了达特茅斯会议，但现在有证据表明会议还有其他的列会者。后来一直做神经网络硬件研究从而躲过 AI 几十年过山车的斯坦福大学电机系教授维德罗（Bernard Widrow）后来回忆他也去了达特茅斯并且在那儿待了一周。麦卡锡原来的计划是两个月闭门研讨，但并非所有人都对那个事那么上心。纽厄尔和司马贺只待了一周。纽厄尔后来回忆说达特茅斯会议对他和司马贺没什么影响。

尽管是“十仙过海”，但给所有人留下最深印象的是纽厄尔和司马贺的报告，他们公布了一款程序“逻辑理论家”（Logic Theorist），这个程序可以证明怀特海和罗素《数学原理》中命题逻辑部分的一个很大子集。司马贺回忆录里说自己学术生涯最重要的两年就是 1955 年和 1956 年。这篇文章后来成了 AI 历史上最重要的文章之一。

值得注意的是，“逻辑理论家”对人工智能后来的一个分支“机器定理证明”的影响并不大。哲学家王浩 1958 年夏天在一台 IBM-704 机上，只用 9 分钟就证明了《数学原理》中一阶逻辑的全部定理。当然《数学原理》中罗列的一阶逻辑定理只是一阶逻辑的一个子集。目前，一阶逻辑的机器定理证明比起 20 世纪 50 年代已有长足进展，但仍然没有高效的办法。毕竟，王浩证明的是一阶逻辑，而“逻辑理论家”只能处理命题逻辑。数学家戴维斯和哲学家普特南合作，沿着王浩的思路进一步提出了戴维斯–普特南（DP）证明过程，后来进一步发展为 DPLL。王浩对“逻辑理论家”一直持鄙视的态度，认为这是一个不专业的东西。王浩在 1983 年被授予定理证明里程碑大奖，被认为是定理证明的开山鼻祖。司马贺在他的回忆录里则对此表示不满，认为王浩的工作抵消了“逻辑理论家”的原

创性，他们的初衷并不是要有效地证明定理，而是研究人的行为。这是后话，见第 2 章“自动定理证明兴衰纪”。

麦卡锡多年后回忆说：他从纽厄尔和司马贺的 IPL 语言中学到了表处理，这成为他后来发明 LISP 的基础。明斯基后来接受采访时说他对纽厄尔和司马贺的“逻辑理论家”印象深刻，因为那是第一个可工作的 AI 程序。但事实上，明斯基在当时为大会写的总结里对“逻辑理论家”只是轻描淡写。麦卡锡和明斯基明显是一伙的，会议是他们发动的，旨在创立一门新学科。但纽厄尔和司马贺却抢了他们的风头。美国 20 世纪 50 年代的学术氛围不免浮躁，这一帮人又都是年轻气盛、野心十足。

4. 会议之后

达特茅斯会议后不久，1956 年 9 月 IRE（后来改名 IEEE）在麻省理工学院召开信息论年会，麦卡锡受邀做一个对一个月前达特茅斯会议的总结报告。这引起了纽厄尔尤其是司马贺的不满，他们认为麦卡锡只能聊，没干货，而达特茅斯会议唯一的干货是纽厄尔和司马贺的程序“逻辑理论家”。打了一圈架，最后纽厄尔和司马贺做了妥协：麦卡锡先做总结报告，但最后还是由纽厄尔和司马贺讲他们的“逻辑理论家”并发表一篇题为“逻辑理论机器”（Logic Theory Machine）的文章。明斯基认为是他的协调起了作用，但纽厄尔晚年则只对香农的邀请有印象，而司马贺的回忆录则说是大会的主席罗森布拉特和司马贺散了很长一圈步才了断。明斯基机敏异常，讲话时带幽默，但在对这段历史的重构中，却给人印象有点太“刁滑”（cynical），原因也不难猜出。研究历史有时必须得全方位，空间或时间上的接近不见得就真实。太接近时，当事人还都活着，还在一个圈子里混，

不方便互相揭短。但在接近生命末期，或者功成名就，或者人之将死，或者对头已死无所顾忌，敞开了说，有时虽有夸张，但一不留神就会流露真话，纽厄尔属于后者。明斯基“刁滑”可能和他身体好有关系，偌大岁数也没不惑，觉得还有好长的路要走。

科学达人戴森（Freeman Dyson）在他的《一面多彩的镜子》一书中借鉴过伯林（Isaiah Berlin）“刺猬与狐狸”的比喻：刺猬是那些构建理论体系的人，而狐狸则是那些解决问题的人。在他眼里，爱因斯坦、哥德尔是刺猬，而费米、冯诺伊曼属狐狸。科学史有时刺猬得势，有时狐狸当道。是不是可以说纽厄尔和司马贺更像刺猬，而麦卡锡和明斯基更像狐狸呢？具体到 AI 的源头和达特茅斯会议，麦卡锡认为他和明斯基是发起人，纽厄尔和司马贺是“外人”，是搅局者。明斯基的解释是纽厄尔和司马贺一开始的出发点是心理学，这与麦卡锡和他本人的背景不符。但在随后的十年里，他本人更多地走向心理学，而纽厄尔和司马贺更靠近 AI，也没什么矛盾。麦卡锡除了和明斯基关系紧密外，和其他 AI 群体的交流并不多，在所谓其他群体中，最有影响的当属卡内基梅隆那一派了。麦卡锡晚年回忆说那时群体之间的沟通主要是通过研究生，研究生就像大佬们的大使。后来斯坦福大学、卡内基梅隆大学、麻省理工学院的学生确实互为教授，门户之见随着时间的推移逐渐被抹平了。

总之，1956 年 IRE 信息论年会是个值得纪念的会议，除了纽厄尔和司马贺发表的那篇文章之外，心理学家米勒（George Miller）发表了“人类记忆和对信息的储存”（Human Memory and the Storage of Information），这是那篇著名的文章“魔力数字七”（The Magic Number Seven）的另一个版本，不知算不算一稿多发。同在此会上，伟大的乔姆斯基则发表了“语言描述的三种模型”（Three Models for the Description of Language），该文证明了有

限状态句法不能表达某类语言，这是乔姆斯基分层的起源，文中引用了还没出版的不朽名著《句法结构》。乔姆斯基当时刚刚到 MIT 现代语言学系（该系后来演变为语言学与哲学系）出任助理教授并在 MIT 电子实验室做机器翻译的研究。尽管乔老爷后来是“反政府斗士”，但有点反讽的是他早期的研究经费都来自美国空军和海军。

从参与者的角度看，大家会认为这次 IRE 的信息论年会比达特茅斯会议更重要，影响也更深远。米勒回忆说，他当时直觉认识到实验心理学、理论语言学、认知过程的计算机模拟，都是一个“大家伙”里面的组成部分。这个所谓的“大家伙”就是现在的人工智能加认知科学吧。

明斯基回忆自己在达特茅斯会议期间，在纸上画了一个几何定理证明器的设计，并手动模拟证明了等腰三角形的一个定理。会后的 1956 年 9 月，IBM 招了新毕业的物理博士格兰特（Herb Gelernter）实现明斯基的几何定理证明器。麦卡锡此时受到纽厄尔和司马贺的影响，建议在 Fortran 里实现表处理语言，作为实现语言。这个项目在 1959 年实现后，IBM 削减了对 AI 的投入，把这个项目砍掉了，理由是 IBM 不想给人以机器可以替代人的印象。IBM 再次资助 AI 是 20 多年后的 1983 年了，现在好像 IBM 百年老店只能靠 AI 系统沃森（Watson）翻身了。

麦卡锡 1958 年离开达特茅斯学院去了 MIT，帮助创立了 MIT 的 MAC 项目。他和明斯基一起领导了 MAC 项目中的 AI 实验室，1962 年他再次跳槽到斯坦福大学。之后明斯基又和佩珀特（Seymour Papert）合作。计算机操作系统里“分时”的概念是由麦卡锡在 MAC 项目中首创的。他回忆说当时机器太少，但等着上机的学生很多，于是就发明了分时系统。按说分时系统的贡献要比麦卡锡后来的 AI 贡献彰显得多，但麦卡锡得图灵奖可不是靠“分时”，这就像爱因斯坦得诺贝尔奖没靠相对论一样。从这个

意义上 AI 有点像哲学：由此衍生出很多问题，而对这些问题的解决产生出许多子学科；一旦这些子学科独立，就不再待见 AI 了。另一个例子是卡内基梅隆大学的微核心操作系统 MACH，其最早的发源是在卡内基梅隆大学的雷蒂（Raj Reddy）搞的分布式传感网络，MACH 领导者拉希德（Rick Rashid）后来加入微软，MACH 变成微软后来操作系统的基础，他本人也变成微软负责技术的决策者之一。

现在计算机科学已成为成熟的学科，每个计算机系大都有三拨人：理论、系统和 AI。20 年前的美国计算机圈子曾有一种说法：理论和系统的人互相看不起，但又同时看不起 AI 的人。AI 这几年火了，但曾几何时，AI 的人是被压迫者。哲学曾经孕育了科学，但一旦问题被确定，就分离成为单独的科学。最新的例子是逻辑学，现在的逻辑学家都在数学系和计算机系，哲学系被彻底空洞化。哲学家丹尼特（Daniel Dennett）曾说：AI 就是哲学。按照明斯基的说法，人工智能就是先锋派的计算机科学。MAC 项目孕育了计算机科学中很多原创的概念。以至于明斯基后来认为 UNIX 系统是落后的东西，因为他们丢掉了很多 Multics 中的精华。

利克莱德（Joseph Licklider）是信息时代的预言家和布道者，他 20 世纪 60 年代初期在美国国防部“先进研究项目局”（ARPA）创办“指挥与控制”（C2）办公室，后来演变为“行为科学及指挥与控制”办公室，最终变成有权有势的“信息科技办公室”（IPTO）。正是利克莱德最早想到了“人机协同”“计算机网络”“未来图书馆”等先进概念。而他的“行为科学”计划也曾资助过监控项目，不知那是不是受到奥威尔的启发。

1968 年，参议院多数党领袖曼斯菲尔德对 ARPA 的资助方向不满，他认为国防部的钱不能被用于军事目的之外，非军事目的的项目应该由美国国家科学基金会 NSF 负责，ARPA 改名 DARPA，更强调“国防”。利克莱

德遂于 1968 年离开 ARPA，去了 MIT 担任 MAC 项目负责人，统筹 MIT 的计算机科学实验室和人工智能实验室。人们认识到利克莱德的贡献太晚了，他于 1990 年过世。计算机科学最重要的实验室之一施乐 PARC 的创始人泰勒（Robert Taylor）曾称利克莱德是 Johnny Appleseed，这是美国 18 世纪到 19 世纪的园丁查普曼（John Chapman）的外号，他把“苹果树”的种子遍撒美国。


利克莱德（1915—1990）

20 世纪 70 年代初期在海尔梅尔（George Heilmeirer）任内，DARPA 大砍 AI 预算。协调政府和 AI 实验室的工作变得头绪繁多，明斯基决定从 AI 实验室退位，让他刚毕业的学生温斯顿（Patrick Winston）接手。

尽管明斯基说他不喜事务性工作，但他的采访和回忆中触及的话题总是和联邦政府的资助有关。温斯顿后来回忆时说，管理一个成功的实验室要管理好三个圈的交集：出资人（主要是政府）、科学上有创建、有国计民生的价值。他试图说服几任 ARPA 的头儿别把 AI 当作一个几年一次的项目，而是长期而独立的一门学科。另外他对比了早期 ARPA 和 NSF 的不同，NSF 是 20 世纪 80 年代才开始资助 AI 研究的，且给钱少，而且都是同行评议制，结果是越有成就的拿的钱越多，但很少会有根本性的原创性贡献，ARPA 早期都是头儿们说了算，好处是如果管事的头儿们品味

好，肯定会支持好东西。这一点也值得一些科技人借鉴：大型项目决策者的品味可以超越“透明计算”吗？

再说回海尔梅尔，他以 AI 不能帮助造武器打仗为理由，削减了对 AI 的大规模经费，但同时却重金资助了隐形飞机和空间武器技术，使美国在相关领域一直保持领先。ARPA 资助的这类项目要是通过同行评议是很难实施的。ARPA 几乎在同时也支持了 ARPANET，后来演变成互联网。有意思的是，海尔梅尔从 ARPA 离任后去了德州仪器（TI）做 CTO，在 TI 却大力提倡 AI。ARPA 对 AI 的资助在克柔克（Steve Crocker）手里才逐步恢复。大家知道克柔克是互联网的先驱之一。再后来的 ARPA 信息技术办公室（IPTO）的负责人中还有图灵奖获得者萨瑟兰（Ivan Edward Sutherland），也对 AI 继续投入。精英制风格的 ARPA，更适合做大型开创性项目，成功取决于少数决策者；而以民主制为基础的 NSF，历来就是小规模资助基础研究。

5. 预测未来：会有奇点吗？

司马贺 1957 年曾预言十年内计算机下棋会击败人。1968 年麦卡锡和象棋大师列维（David Levy）打赌说十年内下棋程序会战胜列维，最后赔了列维两千块。乐观的预言总会给对手留下把柄：德雷弗斯后来每年都拿此事嘲讽 AI，说计算机下下跳棋还行，下象棋连十岁的孩子都干不过。这便宜话一直说到 1997 年，IBM 的下棋程序“深蓝”击败了卡斯帕罗夫。这真是“四十年太久，只争朝夕”啊。在 1995 年卡斯帕罗夫还在批评计算机下棋缺乏悟性（insights），但 1996 年时他已经开始意识到“深蓝”貌似有悟性了。而两年间“深蓝”的计算能力只不过提高了一倍而已。机器

有没有悟性的边界其实就是人的解释能力的极限。量变到质变的临界点就是人的解释能力，人解释不了的东西就有悟性，解释了的东西就没有悟性。司马贺和日本计算机科学家宗像俊则（Toshinori Munakata）合写了篇解气的文章“人工智能的教训”（AI Lessons）登在《ACM 通讯》上。

当然，德雷弗斯们还可以将“计算机仍然不能干什么”加上若干个“仍然”接着批评。明斯基 1968 年在库布里克的电影《2001 太空漫游》的新闻发布会上曾大放厥词说 30 年内机器智能可以和人有一拼，1989 年又预言 20 年可以解决自然语言处理。现在我们恐怕还不能说机器翻译器令人满意吧。过分乐观的另一个原因，照明斯基自己的说法是，一门年轻的学科，一开始都需要一点“过度销售”（excessive salesmanship）。但是过头了不免被人当作狗皮膏药或炼金术。

2006 年，达特茅斯会议 50 周年时，当时的 10 位与会者中有 5 位仙逝，活着的 5 位：摩尔、麦卡锡、明斯基、塞弗里奇和所罗门诺夫在达特茅斯团聚，忆往昔展未来。



2006 年，会议 50 年后，当事人重聚达特茅斯（左起：摩尔、麦卡锡、明斯基、塞弗里奇、所罗门诺夫）

参会人之一霍维茨（Horvitz）现在是微软实验室的头目，他和他老婆拿出一笔钱在斯坦福大学捐助了一个“AI100”[①]的活动：在下面 100 年里各路豪杰聚会，每 5 年出个 AI 进展报告。第一期出版于 2016 年，但里面并无什么干货。

乔姆斯基晚年边做学问边做斗士。2015 年 3 月他和物理学家克劳斯对话时被问及“机器可以思维吗？”，他套用计算机科学家戴客斯特拉（Dijkstra）的说法反问：“潜艇会游泳吗？”如果机器人可以有意识（consciousness）的性质，机器人可以被认为有意识吗？他进一步说“意识”是相对简单的，而“前意识”（preconsciousness）是困难的问题。他把 AI 分成工程的和科学的。工程的，如自动驾驶车等，能做出对人类有用的东西；科学的一面，乔老爷明显不认可。他引用图灵的话：这问题没有讨论的意义（too meaningless to deserve discussion）。当一帮奇点理论的粉丝带着正面的期望采访乔姆斯基时，他却对人工智能这个被他深刻影响过的学科没太当回事，他认为气候和毁灭性武器是比奇点更紧迫的问题。这算有意回避吧。

明斯基在 2012 年接受他的学生、预言家、奇点理论炮制者库兹韦尔的采访时说，他相信奇点的到来，可能就在我们的有生之年。两位“斯基”在麻省理工学院 150 周年纪念会上分在一个小组讨论里，却只打了下太极，并没有针锋相对。明斯基 2016 年 1 月 24 日在波士顿去世，据说为了等奇点，他老人家把自个儿冷冻了。

① AI100 活动在斯坦福有个网站：https://ai100.stanford.edu/。



明斯基和乔姆斯基在麻省理工学院 150 周年纪念会上同室不操戈，并没针锋相对

参考文献指南

人工智能是一门新学科，历史的读物并不多。波登的《认知科学历史》（Boden 2008）和尼尔森的《人工智能探究》（Nilsson 2010）是两本严肃的读物。麦克达克（Pamela McCorduck）曾是费根鲍姆的御用作家，她 1979 年写的《能思考的机器》（*Machines Who Think*）一书，无论是取材还是立意，从今天的角度看都略微过时。尼尔森是人工智能学科的早期参与者，也一直是领导者之一，他多年担任 SRI 的人工智能部门负责人和斯坦福大学计算机系主任，是圈里人。

纽厄尔 1981 年的文章探讨了如何研究人工智能的历史，他总结了人工智能历史中不同思想的对立，他的方法也可以用来研究更广义的计算机科学，甚至可以拓展到不同科学领域和哲学。尽管这是 30 多年前的文章，但今天读来仍有启发。

明尼苏达大学的巴贝奇研究所是专门研究计算机科学历史的机构。主持工作的诺伯格采访了多名计算机科学家，并做了录音。这些被采访的

人中也有不少人工智能学者，例如纽厄尔、麦卡锡、明斯基、温斯顿、布坎南等。听这些人的录音采访和阅读正儿八经的文章完全是两种不同的体验。采访中的语调幽默，包含了很多文章不可能有的微妙细节。除了录音采访，麦卡锡还有个西蒙斯基金会的更正式的视频采访。

雅各布森（Annie Jacobsen）的《五角大楼大脑》（*Pentagon's Brain*）是关于 ARPA 的详实而有趣的历史。从这本书中我们可以看到信息科技一直不是 ARPA 的主打方向，但互联网这个 ARPA 歪打正着的项目却是它最好的投资。

第 2 章

自动定理证明兴衰纪

As a material machine economises the exertion of force, so a symbolic calculus economises the exertion of intelligence ... the more perfect the calculus, the smaller the intelligence compared to the results.

就像机器能省体力一样，符号演算能省脑力。

演算越完美，付出的脑力就越少。

——W. E. Johnson（约翰逊）

Proof is cultivated reasoning.

证明就是讲究的推理。

——Bruno Buchberger（布赫贝格尔）

1. 自动定理证明的起源

数学哲学有三大派：逻辑主义、形式主义以及直觉主义。逻辑主义的代表人物是罗素，主旨是把数学归约到逻辑，这样只要把逻辑问题解决了，之上的数学问题自然就解决了。也就是说，把逻辑玩转了，数学就不算事儿。希尔伯特主导的形式主义是另一派，他的梦想是把数学形式化，数学过程就是把一串符号变成另一串符号。希尔伯特设想，如果能设计一个大一统的算法，那么所有的数学问题都可以由这个算法来解答。这和逻辑主义精神有一定相通之处。哥德尔后来证明这一切是不可能的。机器定

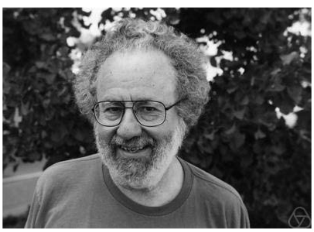
理证明的研究从某种意义上继承了罗素和希尔伯特的思想：用机器来证明和判定那些可以证明和判定的问题。纽厄尔和司马贺的“逻辑理论家”就是早期的机器定理证明程序，他们曾经给罗素写信，期盼能得到伟人的首肯，罗素在回信时说：“我相信演绎逻辑里的所有事，机器都能干。”

自动定理证明起源于逻辑，初衷就是把逻辑演算自动化。逻辑学的源头是亚里士多德的三段论。现代逻辑的奠基人是弗里格（Gottlob Frege）和罗素。弗里格用函数表示谓词，英国人约翰逊（W. E. Johnson）和意大利人皮亚诺使用(x)表示全称，$\exists x$ 表示存在。这样，“人总有一死”，在一阶逻辑中就可表示为(x)Mortal(x)，翻译成普通话就是：“对所有的人 x，x 必有一死。”1935 年德国人根岑（Gerhard Gentzen）仿照∃[exist（存在）的首字母 E 的反写]引入符号∀[all（所有）的首字母的反写]更鲜明地表示全称。根岑的老师是犹太人伯奈斯（Paul Bernays），希尔伯特的合作者。伯奈斯被纳粹赶出德国后，根岑接替老师做了希尔伯特的逻辑助手。根岑是纳粹冲锋队的早期成员，战时积极参与德国导弹 V-2 的研发，1945 年饿死在布拉格的战俘营。他是数理逻辑四大分支之一证明论的奠基者之一，证明论和自动定理证明有着密切关系。

人工智能中符号派的思想源头和理论基础就是定理证明，不懂定理证明就没法深入了解符号派。尽管纽厄尔和司马贺的“逻辑理论家”在人工智能圈子开风气之先，但它还真不是第一个可运行的定理证明程序。这项“第一”的荣誉应归逻辑学家戴维斯（Martin Davis），他 1954 年完成了第一个定理证明程序，所用的机器是普林斯顿高等研究院的一台以冯诺伊曼昵称“强尼”（Johnnie）命名的电子管计算机“大强尼”（JOHNNIAC），而文章则迟至 1957 年才公开发表。

戴维斯是富有成就的数学家和逻辑学家，他比明斯基小一岁，也是

著名的纽约布朗克斯科学高中的学生，那学校已经出了 7 个诺贝尔物理奖得主，一个化学奖得主。戴维斯家境不好，大学时上了被称为“穷人哈佛”的纽约市立学院（CCNY）——那学校不收学费。在那儿他遇见了坡斯特（Emil Post），是坡斯特指引他走上了逻辑之路。他读博士去了普林斯顿大学，导师是丘奇，但戴维斯对启蒙老师坡斯特的感情远胜于丘奇。他出名后，在各种场合为终生不得志的坡斯特鸣不平，他指出坡斯特在 1922 年就获得了和哥德尔 1931 年不完全性定理类似的结果。戴维斯 1957 年写的教科书《可计算性和不可解性》最早系统地介绍了坡斯特的工作，外界才始得知坡斯特的名字。戴维斯 22 岁博士毕业时，23 岁的明斯基才刚从哈佛大学本科毕业，正在来普林斯顿大学读博的路上。戴维斯最重要的贡献是和哲学家普特南（Hilary Putnam）等人解决了希尔伯特第十问题。机器定理证明是他一直感兴趣的副业。

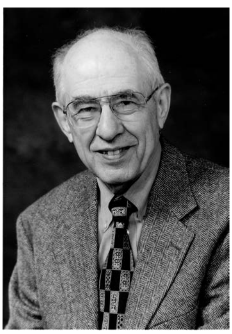


戴维斯（1928—　）

戴维斯的定理证明器实现了普利斯博格算术（Presburger Arithmetic）的判定过程。自然数的一阶理论也称皮亚诺算术，包括自然数的加法和乘法，普利斯博格算术是皮亚诺算术的一个子集，它只有加法没有乘法。皮亚诺算术不可判定，但普利斯博格算术则是可判定的。虽可判定，但其计算复杂性是超指数时间的。这样的算法在“大强尼”上是跑不出什么有意思的结果的，但戴维斯还是为证明器跑出的结果激动不已：两个偶数之和还是偶数——如果没有乘法和除法，偶数定义会稍啰嗦。值得指出的是，

普利斯博格是伟大逻辑学家塔尔斯基在华沙大学时的学生，他在 1928 年证明了一阶自然数加法可判定，之后得了个硕士学位就离开学术界，入了保险行。塔尔斯基门下的另一个著名的数学家学生费佛曼曾说，如果知道普利斯博格工作后来的重要性，真该给他发个博士学位。

后来，戴维斯和普特南继续合作机器定理证明的工作。普特南是 20 世纪最有影响力的美国哲学家之一，他提出的“缸中脑”（Brain in a Vat）是最常被引用的假想实验之一，他和戴维斯等人合作解决希尔伯特第十问题，由此又跨入了数学家行列。他是美国共产党分裂后的组织进步劳工党的活跃党员，晚年对政治失望，68 岁时行了犹太人的成人礼。戴维斯和普特南合作的成果就是影响广泛的戴维斯–普特南（Davis-Putnam，简称 DP）过程，以及后来的 DPLL。



普特南（1926—2016）

戴维斯 2008 年接受美国数学学会杂志 *Notices of AMS* 采访时，只讲了希尔伯特第十问题，以及对连续统问题的看法，压根没提他们的机器定理证明工作。与他们的其他工作相比，定理证明真不算什么。在访谈中，他倒是对哥德尔的哲学观点做了点评。他曾在布朗大学听过哥德尔给数学家讲哲学，哥德尔认为大脑和图灵机不等价。而戴维斯相当直率，他认为大脑就是机器，后来为此还和《皇帝新脑》的作者彭罗斯掐过。20 世纪 80

年代末，戴维斯在南开数学所演讲，我曾问他怎么看当时风风火火的第五代计算机，因为五代机的核心 Prolog 毕竟是定理证明的产物，老头狡猾地呵呵说“我对工业不太懂”，他说的工业泛指人工智能。

纽厄尔和司马贺的“逻辑理论家”被认为是 AI 历史上最重要的原创工作之一，他们的程序可以证明怀特海和罗素《数学原理》第一卷中命题逻辑部分的一个很大的子集。但有一段有意思的插曲，他们把文章满怀期待地投给逻辑学最重要的刊物《符号逻辑杂志》（*Journal of Symbolic Logic*，*JSL*），却惨遭主编克里尼（Stephen Kleene，丘奇的另一位学生）退稿，理由竟然是：把一本过时的逻辑书里的定理用机器重证一遍没啥意思。其实克里尼并非反对机械化的工作，1955 年《符号逻辑杂志》就发表过逻辑学家蒯因（Quine，王浩的老师）的一阶逻辑证明过程的文章，这种理论杂志关注的是有没有原创性，能不能实现倒不是那么重要。蒯因的这篇文章成为后来一批工作的基础。

“逻辑理论家”中首创的“启发式”程序对人工智能和心理学有意义，但逻辑学家却不买账，以至于一些早期的定理证明文章的题目中竟然出现“非启发式程序”的字样，故意恶心纽厄尔和司马贺。王浩甚至称“逻辑理论家”是一个“不专业”的工作，并说“杀鸡焉用宰牛刀，但他们（指纽厄尔和司马贺）拿着宰牛刀也没能把鸡杀了”。这些批评导致做自动定理证明的数学家和逻辑学家在很长一段时间都谨慎地和人工智能圈子保持着若即若离的关系，很多人宁可和计算机理论家混在一起，也不掺和人工智能。

“逻辑理论家”之后，在理论和实践都很有影响的工作是普拉格维茨（Dag Pragwitz）的自然演绎（natural deduction），自然演绎可以追溯到根岑。普拉格维茨自己设计了一个程序语言实现了他的算法，那时不要说编译，

连汇编语言都没有。普拉格维茨的老爸在 1957 年夏天手工把儿子的程序编译成一台瑞典制造的计算机的机器代码。那台机器有 2K 40 位内存（相当于 10K 字节）和 40K 字节的磁鼓外存。于是老普拉格维茨也成了那篇影响深远的文章的作者之一。这项工作不只是自然演绎的开篇，还提出了合一（unification）的概念。

明斯基晚年自诩地回忆自己在达特茅斯会议期间，当场在纸上画了一个几何定理证明器的设计草案，并手动模拟证明了等腰三角形的一个定理。这不禁让我们联想到当下：在草纸上当场写商业计划书并得到巨额风险投资的神话，这一类段子的"始作俑者"是明斯基，虽不知这事能否全信，倒也符合明斯基的个性。达特茅斯会议后的 1956 年 9 月，IBM 招了新毕业的物理学博士格兰特（Herb Gelernter）来实现明斯基的几何定理证明器，1957 年觉得人手不够，又从宾夕法尼亚州立大学招来了逻辑学家吉尔莫（Paul Gilmore）帮忙。为了学习 IBM 704 上的汇编语言，吉尔莫决定先试着实现语义表（Semantic Tableau）方法，最后的结果居然和普拉格维茨异曲同工，但他在一年后才得知普拉格维茨的工作比他要早一年。

1958 年夏天，王浩也在一台 IBM 704 机上实现了一个完全的命题逻辑程序，以及一个一阶逻辑程序。后者只用 9 分钟就证明了《数学原理》中一阶逻辑的全部 150 条定理中的 120 条。到 1959 年夏天，改进版本证明了全部 150 条一阶逻辑以及 200 条命题逻辑定理。例如，《数学原理》中的定理*11.26 被罗素和怀特海写为：

$$*11.26.\ |\text{--}:.(\exists x):(y).\Phi(x,y):\supset:(y):(\exists x).\Phi(x,y)$$

现代教科书的记法是：

$$(\exists x)(\forall y)\Phi(x,y)\supset(\forall y)(\exists x)\Phi(x,y)$$

如果 $\Phi(x, y)$表示"x 和 y 是一双鞋"，这句话可以翻译为"如果存在

一个 x 对所有的 y 都能配成一双鞋的话，那么对所有的 y，都必定存在一个 x 能和它配成一双鞋”。不知道 x 和 y 哪一只是水晶鞋。

王浩在打孔卡片上的记法则是：

11*26/EXAYGXY-AYEXGXY

很明显这是为了节省复杂公式的语法解析（parsing）成本。

当然《数学原理》中罗列的一阶逻辑定理只是一阶逻辑的一个子集。王浩注意到《数学原理》里的一阶逻辑公式都是 AE 形式（即前面是全称量词，后面是存在量词），后来他又继续研究 AEA 的可计算性和复杂性，由此引出了他的学生库克（Stephen Arthur Cook）的 NP 理论——库克 1971 年发表的文章的题目恰好是《定理证明的复杂性》，因此获得 1982 年图灵奖，所以可以公正地说，王浩的定理证明研究孕育了整个理论计算机科学。库克在回忆老师时说：“我很了解王浩的思想和方法，我对 NP 完全问题的结论与他非常相似。图灵和王浩研究谓词逻辑，我研究命题逻辑。”

王浩以哥德尔的权威诠释者和知音名世，但他对哲学、逻辑学、计算机科学的原创贡献却被低估。王浩一共只培养过 5 个学生，都是 20 世纪 60 年代在哈佛大学教数学和逻辑时，之后他离开逻辑学和计算机科学，专心于哲学了。与此对照，他的学生库克则有 34 个学生，库克本人至今仍活跃在本领域的前沿。



哥德尔（1906—1978）与王浩（1921—1995）

1983 年，国际人工智能联合会（IJCAI）授予王浩自动定理证明里程

碑大奖，王浩在致获奖辞时半开玩笑地说因为自己的个性，荣誉经常绕道而行。这和他老师蒯因对他的印象截然不同，蒯因认为王浩一直不满现状，并不断跳槽。王浩回忆起若干年前，他在洛克菲勒大学的顶头上司李德伯格（Joshua Lederberger）校长给他推荐了雷纳特（Douglas Lenat）的博士论文。那是关于用机器进行启发式数学学习的，一段时间内在人工智能界很有影响。但王浩看后觉得雷纳特的东西基础不牢靠，逻辑学家自然有资格随时对计算机科学家说三道四。作为诺贝尔生物奖获得者，李德伯格在成为洛克菲勒大学校长之前是斯坦福大学的遗传系主任，一直在和雷纳特的老师费根鲍姆合作，用人工智能技术解决化学问题，他们合作的成果就是第一个专家系统 Dendral。其实，雷纳特也意识到自己早期工作的问题，他后来走上了一条三十年的不归路——Cyc，用逻辑表达人类的常识，为现今的知识图谱奠定基础、积累经验。这是后话。

王浩的定理证明程序后来成为高级语言的基准程序，麦卡锡的 LISP 早期就一直以王算法的程序作为例子。王浩对“逻辑理论家”的批评后来被哲学家德雷弗斯用来攻击 AI，这也使王浩不爽。尽管王浩的工作被人工智能学界渐忘，但他的逻辑圈朋友都十分认可。

戴维斯的几篇关于定理证明历史的文章题目下面都不忘写一行“纪念王浩”。王浩毕业于神奇的西南联大数学系，曾和杨振宁同屋，二战结束后又到清华大学在金岳霖和王宪钧指导下得了哲学硕士，到哈佛大学后跟随美国最有影响的哲学家蒯因研究逻辑和分析哲学。20 世纪 50 年代，王浩曾一度想回国效力，于是他从哲学转向计算机，因为计算机更加实用，对祖国建设的贡献更加直接。戴维斯证实，王浩在收到他父亲指责他的信之后放弃了回国的念头，他认为他父亲的信是在压力之下写的。关于这段历史见尼克《哲学评书》。王浩没能回国，而机器定理证明则成为王浩计

算机科学转向期间的一个插曲。

2. 罗宾逊和归结原理

1957 年夏季，在塔尔斯基的号召下，一个逻辑学家的大聚会在康奈尔大学举行，会上数学家亚伯拉罕·罗宾逊（Abraham Robinson，数学和逻辑领域有好几个罗宾逊，且互有关联）指出，埃尔布朗（Herbrand）定理可以把一阶逻辑的问题转化为命题逻辑。这激发了大家寻求统一高效的定理证明的实现方法。

英国人阿兰·罗宾逊（John Alan Robinson）1952 年在剑桥得了古典学学位后来到美国，1956 年在普林斯顿大学哲学系亨培尔（Carl Hempel）指导下得了博士，但他的实际导师是年轻的普特南。有意思的是，罗宾逊在读博士时并不知道导师普特南正在和戴维斯合作研究机器定理证明。罗宾逊毕业后先是到杜邦公司研究运筹学，1961 年他谋得在赖斯大学（Rice）哲学系教逻辑的职位，但每年夏天还是到阿贡国家实验室做机器定理证明的研究。后来他索性全职加入了刚成立的阿贡定理证明小组，一位同事是和他同姓的软件工程师乔治·罗宾逊，另一位就是随后成为阿贡定理证明小组的头儿的传奇人物、盲人数学家沃思（Larry Wos）。事实上，是哲学家罗宾逊和码农罗宾逊用蒯因的入门教科书《逻辑方法》教会了数学家沃思逻辑。阿兰·罗宾逊一开始的任务是实现 DP（戴维斯–普特南）过程，这回还是用 IBM 704，但此时已经有了高级语言 Fortran。在实现 DP 的过程中，他发现了对后来定理证明有长远且深刻影响的归结（resolution）原理。有时，一个重大的发现或发明是在深刻体会已有工作的过程中自然出现的。相关文章的发表却因为一名匿名审稿人的疏忽而耽搁，迟至 1965 年

才公开发表在 *JACM* 上。



阿兰·罗宾逊（1930—2016）

阿兰·罗宾逊受到普拉格维茨工作的启发，拓展了普拉格维茨的原始合一算法，发明了归结原理。以前的定理证明技术会用到很多规则，有了归结后，所有的证明推导只要有归结这一条规则就可以了。据后来考证，归结方法在 1937 年就被布莱克（Archie Blake）在其关于布尔代数的博士论文中发现过，而蒯因在 1955 年简化布尔函数时也独立发明过。但无论如何，把合一算法和归结原理结合并应用到一阶逻辑是罗宾逊的原创，这是定理证明中的重要里程碑。

在罗宾逊 1965 年的文章中，一个被忽视的贡献就是包含（subsumption）。虽然归结极大地简化了定理证明，但是它的一个副作用是可以很快生成大量新子句，这和其他定理证明技术是类似的。大部分新子句都是没必要保存的，这就像图书馆时不时要清理旧书，相当一部分书的内容已经被另一部分书所包含。例如我们知道“所有人必死”Mortal(x)，那么当生成“苏格拉底必死”Mortal(Socrates)作为中间子句时，它没必要被存下来，也就是说 Mortal(x)包含了 Mortal(Socrates)。沃思等人在 1991 年纪念罗宾逊六十大寿的文集上撰文指出包含是罗宾逊的另一大贡献，其重要性甚至要超过

归结原理。罗宾逊的贡献不能单纯用归结来总结，他的原创性在于一系列工作的综合，除了归结外，还有合一和包含。

3. 项重写

项重写（term rewriting）其实就是数学哲学中希尔伯特形式主义的精髓：证明就是将一串公式重写成另一串公式。例如，乘法的分配律就是：

$$a(b+c) \rightarrow ab+ac$$

“→”左边的公式被重写成右边的公式。重写规则就是单向的方程。方程式（等式）是一阶逻辑的子集，即只有一个谓词 EQUAL。仿照哥德尔的一阶逻辑完全性证明，美国数学家伯克霍夫（George David Birkhoff）证明了方程逻辑是完全的。考虑到方程在数学中的广泛使用，方程逻辑一直是逻辑中一个活跃的领域。数学活动都可以被看作对公式的重写，方程式求解的机械化的重大突破是算法大师高德纳（Donald Knuth）和他的学生本迪克斯（Peter Bendix）做出的。他们的工作为项重写提供了坚实的基础。其工作的重要性不亚于罗宾逊的归结原理。如果说罗宾逊 20 世纪 60 年代奠定了逻辑主义定理证明，高德纳和本迪克斯则在 1970 年开启了形式主义证明方法。

从某种意义上说，丘奇的 λ 演算就是项重写。针对谓词是等词的情况，沃思等人则提出了 paramodulation 和 demodulation（项重写的一种简单变种）。paramodulation 后来被推广为 superposition，成为现代定理证明器的理论和实践基础。

4. 阿贡小组和马库恩

芝加哥大学的实验物理学家费米在曼哈顿计划的早期负责核反应堆。战后，美国以费米实验室为基础建立了第一个美国国家实验室阿贡（Argonne），隶属于美国能源部。阿贡国家实验室的一个分部是应用数学和科学计算，阿贡定理证明小组就在这个分部。阿贡小组对机器定理证明的贡献是决定性的和全方位的。罗宾逊就是在阿贡工作时提出了归结原理。阿贡为归结原理的实现提供了一整套生态环境。小组的头儿沃思尽管是盲人，却是数学天才，他 14 岁就赢得芝加哥大学的奖学金，但他等到 17 岁生活能自理后才入学。

计算机科学的大部分分支都要求不只是理论好，还得能构建系统。阿贡小组的精神就是实战。小组的另一员干将欧文白克（Overback）曾和摩尔（Moore）讨论理论和实际之争，摩尔说："编程就像做爱，没法叫别人替你干。"阿贡小组除了 paramodulation，还提出了支持集（Set of Support，简称 SoS）的概念。在证明定理中，通常会把给定的公理和待证的结论分开，一般不会在公理内互相归结，这样可以极大提高定理证明的效率。支持集就是公理之外的子句集。其实人在证明定理时也是如此。沃思证明支持集是完备的。

除了技术方面，沃思为定理证明做出的最大贡献就是把马库恩（William McCune）招到阿贡实验室。马库恩遂成为 20 世纪 90 年代定理证明领域快速发展的重要推动者。他理论实践两手都硬。在加入阿贡实验室后，用 C 语言写了 Otter 定理证明器，Otter 实现了当时定理证明里最先进的所有技术。美国人工智能的主要语言是 LISP，马库恩和人开玩笑说他不会写 LISP，只会写 C。



马库恩（1953—2011）

包含测试是定理证明中最花时间的，很多证明器要花超过 95%的时间做包含测试。相比正向包含，反向包含测试有点得不偿失，有些定理证明器选择不做反向测试。马库恩是最早把项索引（term indexing）引入到机器定理证明器的。Otter 主要用到了两种项索引，一种是路径索引（path indexing），另一种是马库恩自己发明的差别树索引（discrimination tree indexing）。差别树索引对正向包含极为有用，正向包含测试是比反向包含更耗时的操作，差别树极大地提高了证明的效率。

马库恩还利用 Otter 的模块开发了另一款专门证明方程的证明器 EQP。1996 年 10 月 10 日，马库恩用 EQP 证明了罗宾斯（Robbins）猜想。这是数学家罗宾斯 1933 年提出的一个关于布尔代数的猜想，60 多年来从未被证明。EQP 在一台 486 机上跑了 13 天给出了证明，之后又在一台 IBM RS/6000 工作站上跑了 7 天进行了验证。《纽约时报》马上报道了这一里程碑事件。马库恩在第一时间打电话给已经 81 岁还在罗格斯大学（Rutgers）任教的罗宾斯，说一开始他们认为罗宾斯猜想的第三公理可能是个印刷错误，罗宾斯听了颇为得意。1957 年司马贺曾预言计算机将在十年内击败

人类象棋冠军时，还预言十年内机器能证明人没能证明的定理。实际上这花了 39 年。一年后的 1997 年，IBM“深蓝”击败了卡斯帕罗夫。

5. 符号派的内部矛盾：问答系统和归结原理的失落

大多数定理证明器都使用反证法，也就是把给定要证明的命题的否命题输入系统，得出矛盾（即空子句），从而证明原命题。定理证明一般只能回答“是”或“否”的问题。有时我们需要证明结果提供更多信息。假设我们面对一个定理证明系统，想知道它是否包含这样的事实“谁是玛丽的丈夫？”那么我们可以问：

~ Husband(*x*, Mary).

也就是问我们想要知道答案的问题的否命题。如果定理证明系统里已经包含了另一条命题：

Husband(John, Mary).

那事情很简单，两个子句归结，产生空子句。我们从而知道“玛丽是有丈夫的”，但我们仍然不知道他是谁。格林（Green）在 1969 年的斯坦福大学博士论文里想到了一个简单的解决办法——在被要求证明的子句里附加一个系统谓词 ANS，那么我们可以问：

~ Husband(*x*, Mary) | ANS(*x*).

归结后得到 ANS（John）。由于 ANS 是系统谓词，这个结果仍然是空子句。但 ANS 中包含了我们想要的结果：玛丽的丈夫是约翰。

ANS 这个系统谓词已经在 Otter 这样的定理证明器中得到实现。在 Otter 中系统谓词都以$开头，$ANS 的作用其实就是跟踪变量的值。定理

得证时（即子句变空时），$ANS 中变量的值就是问题的答案。

纽厄尔在总结 AI 历史时用了“路线斗争”的方法，即在任何时刻，每种方法都有个对立面。与定理证明密切相关的路线斗争涉及定理证明与问题解决（Theorem Proving vs Problem Solving，1965）以及过程表达与陈述性表达（Procedural vs Declarative Representation，1970-1980）。在这两场争斗的美国主战场，定理证明和陈述性表达的通用性被认为是低效的，没法用来解决实际问题。定理证明的共同体从此分化，分为“纯的”和“不纯的”两派。“不纯的”一派认为必须引入过程表示，而“纯的”一派则认为引入过程知识是“作弊”，智能程序本身也得有智能，只要把问题陈述出来，定理证明程序就应该智能地工作，不必依靠编程者的过程知识。于是，在 PLANNER、QA3/QA4 等带过程表示的系统和语言被开发出来以后，除了阿贡小组在坚持外，“纯的”定理证明学者在美国的日子并不好过。实际上，把函数恰当地引入 ANS 谓词中，PLANNER 的问题都可转化为一阶逻辑的问题，不多也不少。

纽厄尔总结的另一对对立面就是逻辑与心理（Logic vs Psychologic）。他把这段路线斗争的时间区间定在 1910 年到 1945 年（1910 年怀特海和罗素的《数学原理》出版，1945 年麦卡洛克和皮茨的神经网络文章诞生）。这大概与王浩曾不留情面地批评过他和司马贺有关，他们的工作不被逻辑学家认可，当然只能是心理派，而在文中他似乎也认为逻辑与心理之争并没有完结。

符号派的掌权者多是心理派，没受过科班逻辑训练，而理论家又时常会嘲讽逻辑不好的实干家是民科。归结原理导致的组合爆炸仍然需要启发式方法的帮助。于是，心理派的人又开始了对逻辑派的攻击，费根鲍姆批评罗宾逊把人工智能引入陷阱。罗宾逊对定理证明的停滞也束手

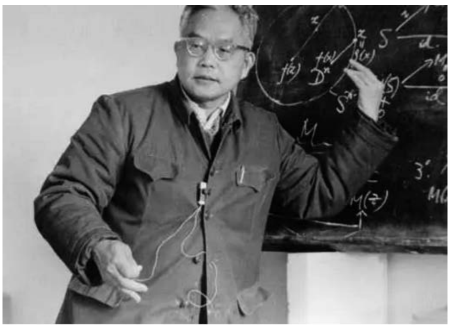
无策，被迫从《人工智能杂志》编委会辞职，最后回归逻辑研究。这也算是费根鲍姆替他老师纽厄尔和司马贺对王浩早年对他们尖刻批评的报复吧。而 20 世纪 80 年代初期阿贡小组核心成员温克（Steve Winker）在给一份著名的 AI 杂志投稿时，资深编辑都懒得把稿件送给匿名评审，只是回信说“也许 *JACM* 还在发这类东西吧，尽管我也不知道为啥（他们还发）”。*JACM* 是美国计算机学会会刊，以发理论文章著称。联想到早些时候，纽厄尔和司马贺“逻辑理论家”的论文惨遭《符号逻辑杂志》主编克里尼退稿，我们只好感叹“冤冤相报何时了”。欧文白克愤怒地回忆某知名人工智能新秀对归结原理的不屑，那位后来变成权威的人士批评基于归结的定理证明器不能解决实际问题。欧文白克多年后说权威这辈子都没能干出一个可以工作的系统，而马库恩的 EQP 已经证明了罗宾斯猜想。

归结的简单性引起了人们的重视，但它同样有组合爆炸问题，在人们对如何驾驭归结没有理论和经验的共识之前，整个领域早已改朝换代，问题解决与否，已经无人关心。20 世纪 70 年代后，阿贡小组依然是美国定理证明的堡垒，但整个学科的研究重心逐渐从美国转向欧洲，特别是英国，在那里，逻辑程序应运而生，成为后来日本五代机的理论基础。

6. 几何定理证明与计算机代数

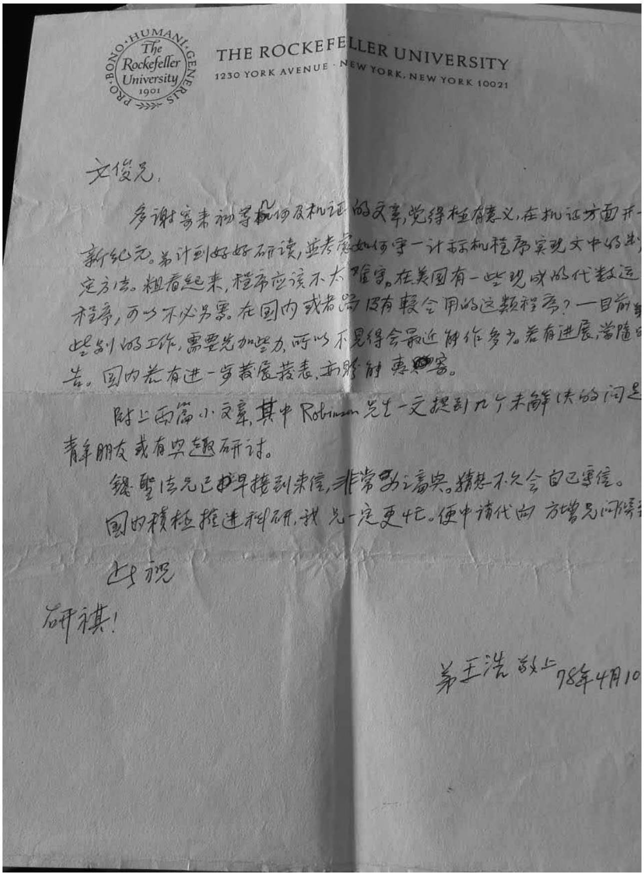
哥德尔证明一阶整数（算术）是不可判定的，但几乎在同时塔尔斯基则证明一阶实数（初等几何和代数）是可判定的。塔尔斯基的证明结果直到二战后在美国成书才得到更为广泛的流传。塔尔斯基一直自认为他应该是和哥德尔比肩的逻辑学家。冯诺伊曼在得知哥德尔不完全性结果时曾

半开玩笑地说，要是哥德尔的结果晚出来几天，他说不定就能证明一阶逻辑是完全的呢。塔尔斯基的结果意味着可以存在算法能对所有初等几何和代数问题给出证明。塔尔斯基的原始算法是超指数的，在被后人多次改进之后仍然很难被当作通用算法。数学家吴文俊在研究中国数学史时，受到启发，针对某一大类的初等几何问题给出了高效的算法。后来吴文俊的方法还被他推广到一类微分几何问题上。



吴文俊（1919—2017）

王浩在得知吴文俊的结果后马上给吴文俊写信，建议他利用已有的代数包，甚至考虑自己动手写个程序实现吴的方法。其实，后来塔尔斯基算法的各种改进都被吸纳进数学程序包 Mathematica 中了。不严格地说，如果哥德尔系逻辑的机器体现是图灵机的话，塔尔斯基系逻辑的机器体现就是实数模型。王浩和吴文俊的通信大概是哥德尔系定理证明和塔尔斯基系定理证明为数不多的交流。


THE ROCKEFELLER UNIVERSITY
1230 YORK AVENUE · NEW YORK, NEW YORK 10021

文俊兄，

多谢寄来初等几何及机证的文章，觉得极有意义，在机证方面开一
新纪元。弟计划好好研读，並考虑如何写一计算机程序实现文中的判
定方法。粗看起来，程序应该不太难写。在美国有一些现成的代数运
程序，可以不必另写。在国内或者尚没有较合用的这类程序？一目前
些别的工作，需要先加些力，所以不见得会最近能作多少。若有进展，当随
告。国内若有进一步发展发表，亦盼能惠寄。

附上两篇小文章，其中 Robinson 先生一文提到九个未解决的问题
青年朋友或有兴趣研讨。

钱圣陶先生已早接到来信，非常高兴。猜想不久会自己写信。

国内积极推进科研，我兄一定更忙。便中请代向方增兄问候

此祝

研祺！

弟王浩敬上　78年4月10

王浩 1978 年 4 月 10 日写给吴文俊的信

目前，基于逻辑的定理证明器最适合解决代数问题。而几何定理证明器却又都是基于代数的。王浩是逻辑系定理证明的先驱，吴文俊则开几何系定理证明的风气之先。哥德尔定理和塔尔斯基定理在人工智能问题上各有蕴意，是为后话。有意思的是，塔尔斯基对机器定理证明的结果不感兴趣。在 1957 年康奈尔大学的逻辑学家大会上他听到戴维斯讲实现他学生普利斯博格的判定过程时如此，甚至在后来听到科林斯和科恩（就是证

明了连续统独立性的那位，他非常看不起逻辑，但不曾想当年居然也干过这种脏活累活）改进了他自己的结果时，也无动于衷。这让科林斯很沮丧。

吴文俊是数学家中的人精。"文革"期间，他在北京无线电一厂下放，那是家计算机厂，他开始对计算机感兴趣。他最早和计算机相关的论文是讲怎么利用拓扑学给计算机电路布线的。数学家学会了计算机编程，试试机器证明是最自然的。一开始的算法都是手工推演，1977 年大年初一，吴文俊取得了突破。同年，他的文章"初等几何判定问题与机器证明"发表在《中国科学》上。吴文俊声称他的成果是在研究中国数学史时受到的启发。

周咸青（Chou Shang-ching）1978 年考到中科院计算所，师从唐稚松，他正好赶上吴文俊开几何定理证明的课，就去旁听了。那时吴文俊的书《几何定理机器证明的基本原理》还没正式出版，但所有旁听生都已拿到手稿。周咸青后来到得克萨斯大学奥斯汀分校留学，师从波尔（Boyer）和布莱索（Woody Bledsoe）。他们从周处听说吴文俊的工作很是兴奋，这两位虽都是逻辑系定理证明的大咖，但他们有足够的宽容，让周咸青对自己的博士论文题目自作主张，周的论文基本就是吴方法的实现。奥斯汀分校的硬件设备当然比吴文俊的环境好多了，周取得的成果自然也更加丰富。吴文俊的名字由此传向自动定理证明界。

"文革"后的中科院数学所派系纷杂，关肇直和华罗庚不和，吴文俊和吴新谋不和。乘着华罗庚在欧洲访问，关肇直在中科院副院长钱三强的支持下，迅速成立中科院系统科学所，吴文俊毫不犹豫地跟随关肇直到了系统所。等华罗庚从欧洲回来，生米已然成了熟饭，华罗庚单约吴文俊请他回数学所，被吴婉拒。吴文俊虽和华、关都保持友好关系，但关肇直更理解吴文俊。20 世纪 50 年代，华罗庚曾在数学所建立逻辑组，并邀胡世

华领衔。后几经周折，逻辑组先被分派到计算所，后来软件所从计算所分家时，逻辑组再到软件所。直到 1998 年，原来分出去的几个所才又重新集结在“数学与系统科学研究院”的名下，吴文俊的“数学机械化研究中心”也成了这个新研究院的重要组成部分。研究院的英文名字居然是 Academy of Mathematics and System Science，不知道这段历史的，大概会以为他们是和中科院同级的 Academy 吧。

1979 年，吴文俊的工作得到杨振宁的关注，当时的科学院副院长李昌和关肇直都大力支持吴文俊，并为他申请到两万五千美元的外汇到美国购买一台家用电脑，以实现他的吴方法。1979 年，普林斯顿高等研究院邀请吴文俊和陈景润访美。吴文俊在普林斯顿讲了拓扑、中国数学史和机器定理证明。普林斯顿的人对机器证明不感兴趣，吴文俊有点失望。吴文俊还顺道访问了加州大学伯克利分校，在那儿见到了菲尔茨奖得主斯梅尔（Steve Smale）。斯梅尔是实数计算模型 BSS 的原创者之一，他高度评价了吴文俊的工作，这令吴文俊欣慰。斯梅尔出名也是靠拓扑，但晚年转到计算理论，和吴文俊惺惺相惜吧。吴文俊此次访美的重要一站是去王浩所在的洛克菲勒大学。吴文俊的工作在定理证明界迅速引起重视，王浩是另一个关键推手。在波尔和项洁的推荐下，吴文俊 1997 年获得第四届埃尔布朗奖，这是定理证明领域的最高奖项。

吴文俊的长寿也体现在他的学术生命上。1979 年吴文俊 60 岁高龄开始学习计算机编程语言，先是 BASIC，后是 Algol，再后是 Fortran。他在那台两万五千美元的家用电脑上不断取得新的成果。后来系统所的硬件设施改进，吴文俊相当一段时间都是上机时间最长的。

在定理证明的早期，研究者追求“一招鲜吃遍天”，就是找到一个超级算法能证明所有问题，最典型的例子是归结法和后来的 superposition。

王浩不认可这种思路，他认为他自己的早期工作和吴文俊的方法都表明最有效的方法是先找对一个相对可控的子领域，然后针对这个子领域的特性，找到有效的算法。

吴文俊后来喜欢用一个比“机器定理证明”更大的词儿“数学机械化”（Mechanization of Mathematics）来描述他的工作，这个说法借自王浩（Wang 1960）。吴文俊的哲学思想是典型的数学家思路，这和逻辑学家不同。数学家把机器定理证明当作工具，而逻辑学家则把机器定理证明当作目标。数学家主要看是否有用，而逻辑学家则看是否纯粹。吴老在一次讲座中讲计算机和数学机械化，引用维纳的说法：“人脑贬值，至少人脑所起的较简单、较具常规性质的判断作用，将贬值。”笛卡儿认为代数使得数学机械化，因而使得思考和计算步骤变得容易，无需花很大脑力。小学算术很难的东西，初中代数立个方程马上就解了。每一次数学的突破，往往以脑力劳动的机械化来体现。

杨振宁曾说他最重要的成就是提高了中国人的自信。陈省身、华罗庚、杨振宁、李政道那一批人是最早为人类文明做出贡献的中国人。那个不长的名单里还应该有王浩和吴文俊。

7. 定理证明系统和竞赛

迈阿密大学的萨特克里夫（Geoff Sutfliffe）每年都组织机器定理证明大赛 CASC（CADE ATP System Competition），主要是针对命题逻辑和一阶逻辑证明器。萨特克里夫还维护一个 TPTP（Thousands of Problems for Theorem Provers）的网站，这里有大量的定理证明程序的基准测试例题。2000 年之前的大赛中，Otter 几乎在所有的领域都力拔头筹。2005 年后曼

彻斯特大学的“吸血鬼”（Vampire）后来居上，至今一直维持领先状况。

除了命题逻辑和一阶逻辑外，高阶逻辑（主要是类型论）的自动定理证明在欧洲各研究机构一直有人研究。每年“高阶逻辑定理证明”的会议录都被收入斯普林格出版社（Springer）的 LNCS 系列。

如果说 Otter 和“吸血鬼”代表了逻辑主义定理证明的高峰，形式主义定理证明器中最广为人知的当属波尔–摩尔证明器及其一系列后续变种。波尔–摩尔证明器的核心是数学归纳法和项重写。波尔–摩尔证明器被用来证明软件和硬件的正确性。有意思的是，他们的另一项工作波尔–摩尔字符串匹配算法（Boyer-Moore string search algorithm）的起因也是定理证明。他们最早用来实现定理证明器的语言是 InterLisp，但后来迅速发现 InterLisp 内带的朴素字符串搜索太慢，他们的算法比朴素算法快了 50 倍，目前仍是最快的字符串匹配算法，几乎在所有的语言和软件包里都有实现。他们后来又用自己的定理证明器来证明自己的字符串算法的正确性。在 2016 年 5 月的维基百科里，Boyer-Moore string search algorithm 词条被引用 61 次，而 Boyer-Moore theorem prover（即 Nqthm）词条只被引用 11 次，真是无心插柳啊。

计算机代数的早期研究者都出自明斯基门下，最有名的程序是 MACSYMA。Mathematica 的作者沃尔弗拉姆（Stephen Wolfram）早期也是 MACSYMA 的用户。现在 Mathematica 也整合了许多定理证明算法，也有人利用 Mathematica 作为定理证明的框架和壳子。其实在马库恩用自己的 EQP 证明了罗宾斯猜想后，就有人用 Mathematica 来验证。

还有一类定理证明器，其功能主要是用来作为验证器（proof checker）。荷兰的德布罗金（de Brujin）的 AUTOMATH 是早期著名的证明验证器。朗道著名的《分析基础》教科书整个被翻译成 AUTOMATH 的形式语言，

形式化证明的长度是原书的 10 倍。

物理学怪才兼企业家弗雷德金（Edward Fredkin）曾为计算机下棋设立过奖项。但不大为人所知的是，他为机器定理证明也设立过一个奖项，分三等，第三等是当前成果奖（Current），1983 年沃思和温克获奖，1991 年波尔和摩尔获奖。二等奖是里程碑奖，1983 年给了王浩，1984 年给了罗宾逊，1991 年给了布莱索。一等奖被称为莱布尼茨奖（注意，这个和德国的那个莱布尼茨奖不同），一次也没发出过，因为条件是“不仅够格在数学杂志上发表，还要够格评选美国数学会的 Cole 奖或 Veblen 奖，甚至菲尔茨奖”。弗雷德金为计算机下棋设定的几个奖项有明确的标准：战胜特级大师，战胜当前世界冠军。按此标准就不难理解为什么马库恩的罗宾斯猜想尚不够格一等奖。沃思认为马库恩应该很接近了。但该奖的评委、哈佛数学家芒福德（David Mumford）想都不想地说：“现在不行，一百年都够呛（Not now, not 100 years from now）。”（Horgan 1993）随着定理证明事业的凋零，该奖后来也悄无声息地撤销了。也有乐观派，例如离散数学家格雷汉姆（Ron Graham[①]）认为在证明定理上计算机超过人是迟早的事，人脑毕竟是生物进化的产物，天生的目的不是用来证明定理的。吴文俊也算乐观派。

8. 哲学问题

定理证明的过程，都是一个归约（reduce）的过程，无论是逻辑派的（即把数学问题归约到更基本的逻辑问题），还是形式派的（即用一套规则不断地变换给定的公式直到显性的形式出现）。自动定理证明研究这个数

① 他的太太是另一位成果丰富的离散数学家金芳蓉（Fan Chung）。

学过程的全自动化。但作为人的辅助工具，有时证明过程是人机互动的。波尔–摩尔证明器就允许人在证明过程中予以干预，例如增加一条引理等，尽管整个过程还是机器主导的。

1976 年阿佩尔（Kenneth Appel）和黑肯（Wolfgang Haken）借助计算机的帮助证明了四色定理，他们把四色定理分成几种情况，其中的一些情况可以借助计算机枚举的方法予以证实或排查。这代表了另一类人机交互，即以人为主，计算机为辅的证明，也称为计算机辅助证明（computer assisted proof）。四色定理证明后，一些哲学家开始挑刺儿，王浩和普特南的学生铁木钦科（Thomas Tymoczko）提出了这样的问题：四色定理算被证明了吗？谁看见了这个证明过程？这听起来有点像乔姆斯基对统计派机器翻译的批评：有黑箱子的理解不能算理解，有黑箱子的证明也不能算证明。传统的数学实践遵循共同体过程，也就是一个数学家提出证明，然后一堆同一共同体的专家来验证，如果验证通过，定理成立，可能后面还伴随证明的改进。费马大定理的证明、庞加莱猜想的证明和张益唐的证明，都是这个套路。不独铁木钦科，很多数学家也持类似态度，毕竟这涉及自己饭碗的合法性。

有些机器证明的定理本身并不长，例如罗宾斯猜想，人可以看得过来。而有些机器证明（无论是全自动的还是计算机辅助的）太长，人根本看不过来，例如布尔–勾股数问题（Boolean Pythagorean Triple Problem）的证明（2016 年 5 月）一共有 200TB，200TB 的 3D 纯高清视频这辈子恐怕都看不过来，更甭说定理了。那怎么才算是证明了定理呢？我们退一步来看，如果那 200TB 文件可以用一个可被信任的计算机程序验证一遍，是不是就算是证明了呢？罗宾斯猜想的证明就曾用 Mathematica 验证过，而 AUTOMATH 本身就是一个验证系统。对全自动的定理证明，验证过程更

容易机械化，而计算机辅助证明可能五花八门，很难有一个统一的过程。

无论如何，数学共同体的实践标准在变：从数学家之间互相核实到数学家信任的程序之间互相核实。也难怪传统的数学家在抱怨：数学变成了有成本的实验科学。英国华威大学（Warwick）的数学教授爱波斯坦（David Epstein）办了本实验数学的杂志（*Journal of Experimental Mathematics*），被他的同行讥讽说应该叫“未证明定理杂志”。爱波斯坦不服，他指出前辈如高斯也利用过实验证明（Horgan 1993）。

其实那些典型的物理科学，例如物理、化学和生物学，是以实验为本的，可重复性（reproducibility）是检验真理的标准之一。只不过在当下，可重复性的成本太高。现在《自然》杂志上发表的那些生物学论文，又有多少是被重复过的？我们可以怀疑作者的诚信，我们也可以感慨，当下的数学变得越来越实验，而生物学可能变得越来越后现代了。可不是嘛，无论是唯心或唯理的数学，还是唯物或经验的实验科学，最终都成了共同体式的实用主义。

吴文俊和芒福德联合得了 2006 年的邵逸夫数学奖。对他们的得奖评语最后都有一句定论，大意是他俩都是从纯数学的分支拓扑最后转到和计算机科学相关的研究，这为数学家的未来行为模式提供了典范。吴文俊声称他的几何定理证明的思路来自中国古代数学思想，他认为丝绸之路使得中国和中亚交流密切，花剌子模的数学思想也必定和中国古代数学思想互有渗透。这可能有点拔高中国数学了。吴文俊曾留学法国，法国的数学家素有关心数学史的传统，和吴文俊相识的人中阿达玛（Jacques Solomon Hadamard）①和布尔巴基学派的丢东聂（Jean Alexandre Eugène Dieudonné）也都是数学史家。吴文俊的结论是中国数学是巴比伦式的而不是希腊式

① Hadamard 也译为“哈达玛”。

的，巴比伦数学讲究计算，而希腊数学讲究公理。

自动定理证明依靠的工具是计算机，正是计算模糊了理性和经验的边界。可以登高一步说：计算是知识演化的基础，计算也是知识民主化的工具。阿达玛曾经研究过数学的心理学，我们无法套用他的很多老套说法。但观察和计算机科学相关的数学家：机器定理证明的开拓者之一沃思是盲人，他专攻逻辑。而麦卡锡尽管在人工智能中走了逻辑这一条路，但他在普林斯顿大学的博士导师却是失去双手的拓扑学家莱夫谢茨。数学家某些感官的缺失会影响到他们不同的数学能力吗？这对他们选择不同的主攻方向会有影响吗？有人说数学家都喜欢艺术尤其音乐，我们不知道是逻辑学家更喜欢音乐还是拓扑学家更喜欢音乐。

9. 现状

阿贡实验室的定理证明小组 2006 年被裁掉了，这大概算是符号派低潮的标志性事件，一个时代结束了。对定理证明做出巨大贡献的马库恩失业了，他那时还不到 53 岁，一年后他走投无路去了新墨西哥大学担任没有终身教职的研究教授。他离开阿贡后无法对 Otter 做大的改进，Prover9 是他的新证明器的构想，但没时间得到完美的实现，Otter/Prover9 的性能遂被其他团队逐渐超越。目前最领先的一阶逻辑定理证明器是曼彻斯特大学的“吸血鬼”。2011 年马库恩因急症去世。他在阿贡的前老板沃思深情地写了一篇纪念文章，沃思提到当年在阿贡时，马库恩的狗得了癌症，兽医给了马库恩两个选择：化疗或安乐死，他选择了后者。

有些领域，一开始就把百分之八十的容易问题都解决了，而后就一直很难，进展很慢，少有突破。人工智能就是这样，定理证明尤其如此。

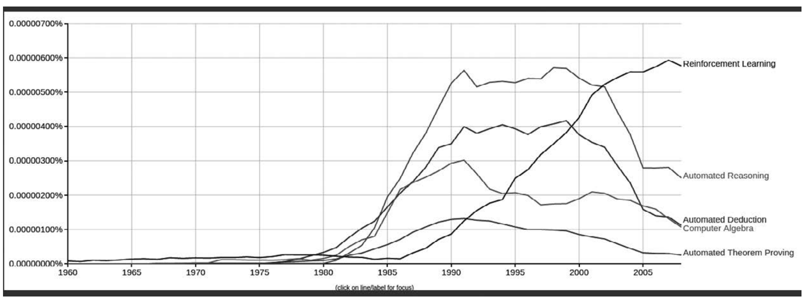
深度学习领域近来的进步更多得益于硬件。而定理证明，即使是硬件再发达很多，也还没有看到曙光。

定理证明是极端的符号派。所有符号派的人工智能技术的基础都是定理证明，如专家系统、知识表示和知识库（甚至数据库）。专家系统的很多术语都是重新包装过的定理证明术语。例如“知识库”就是“公理集合”，“规则库”就是“支持集”，“推理引擎”更是直接照搬。当下流行的（或马上要流行的）知识图谱的基础也是定理证明技术——知识表示的理论“描述逻辑”就是被约束的一阶逻辑的子集。

按照数学哲学的分类，如果说基于逻辑的定理证明技术是逻辑主义的延伸的话，以项重写技术为基础的计算机代数无疑是形式主义的精神遗产。定理证明领域的名字也经历了有趣的演化。最早都叫机器定理证明（Mechanical Theorem Proving），后来改叫自动定理证明（Automatic Theorem Proving），再后来叫自动演绎（Automated Deduction），目前都叫自动推理（Automated Reasoning）。原因很简单，演绎（deduction）只是推理的一种，现在归纳（induction）、溯因（abduction）也都算成推理了。贝叶斯推理，可以叫 Bayesian Logic，或 Bayesian Inference，也可以叫 Bayesian Reasoning。最早的定理证明国际会议（The International Conference of Automated Deduction，简称 CADE）开始是每两年一次，1996 年之后改为每年一次，但奇数年都是自己开，而偶数年是和其他几家计算机科学相关的逻辑会议一起，叫 IJCAR（International Joint Conference of Automated Reasoning）。这促成了自动定理证明圈子和人工智能圈子的融合。定理证明的几个主要会议的会议记录原来都被斯普林格出版社收入到计算机科学讲义（LNCS）系列，现在也都归到人工智能讲义（LNAI）系列了。

如果我们用谷歌 Ngram 来看看 1960 年到 2008 年间的定理证明发展会

觉得很有意思。“自动定理证明”（Automated Theorem Proving）明显在逐渐被“自动演绎”（Automated Deduction）和“自动推理”（Automated Reasoning）所取代，而“计算机代数”和“自动演绎”的流行度差不多。即使是一个很偏门（至少是曾经很偏门）的“强化学习”（Reinforcement Learning），在 2000 年后也要远比“自动推理”流行得多。Ngram 目前只开放到 2008 年，那时还没有现在已经如日中天的“深度学习”。



自动定理证明与强化学习

有意思的是，最近有人开始把深度神经网络应用到定理证明中。一阶逻辑证明中的一个难题是如何从已经证明的子句集合中选取下一个子句，这会极大地影响定理证明器的后续表现。一个谷歌团队的实验（Loos 2017）表明，简单的卷积网络可以帮助定理证明器挑选子句从而提高性能。

10. 结语

王浩曾经抱怨数学家不把逻辑学家当回事。曾几何时，王浩建立的洛克菲勒大学王实验室（Wang Lab）[1]鼎盛时期网罗了当时逻辑学界的所

① 洛克菲勒大学是医学与生物学重镇，学校只培养博士，不设系，每个教授有自己的实验室。

有重要人物（如克里普克、DA 马丁等），俨然是逻辑学的中心，连王浩的老师蒯因也一度想加盟。但在 20 世纪 80 年代，洛克菲勒大学因为经费原因裁减了所有唯心的实验室（包括数学和逻辑学），逻辑实验室只剩王浩一个光竿司令。

按照知识的食物链，生物学家不敢怠慢化学家，化学家不敢怠慢物理学家，物理学家不敢怠慢数学家。乔姆斯基也说过类似的话：物理不好解的问题就升级到化学，化学解不了的就再升级到生物学。但到数学家这儿却不灵了。数学家背上码了一摞乌龟，但他们却说我们肚子底下啥也没有。

维特根斯坦曾有言："逻辑似乎处于一切科学的底部——因为逻辑的研究探索一切事物的本质。"但数学家不觉得他们非得趴在逻辑学家的背上。自动定理证明的状况与此相关，数学家没觉得这玩意儿有用，人工智能的两派人马都不待见。再过几年，可能就没什么人能教"自动定理证明"的课了。

哈尔莫斯（Paul Halmos）是数学家，但也曾涉猎逻辑，还出过一本《朴素集合论》（*Naive Set Theory*），是同类书中销量最好的。哈尔莫斯在自传里拿逻辑开玩笑，说即使有人证明了黎曼猜想是不可判定的（哥德尔就是这么猜测的），数学家睡一觉，第二天起来还是该干嘛干嘛，仍不睬逻辑。他还说数学家才不关心量词顺序是∀∃还是∃∀，这明显是在轻浮地调侃王浩，他也许不知道现在 NP 问题是克雷研究所最关心的千禧年数学问题。哈尔莫斯是 1957 年康奈尔大学逻辑大会的组织者之一，他的自传不无自嘲地引用大会文集结束语的打油诗：

如果你觉得你的文章内容空洞，

那就使用一阶函数演算。

然后它就成了逻辑，
似乎施了魔法，
平常稀松的东西，
就会受到奇迹显现般的欢呼。

所有定理证明系统（无论是逻辑主义的还是形式主义的）的一个致命问题是它们多是独立的，很少和其他数学工具结合，结果必然是只能是玩具系统，而不具实用性。就像早期的计算尺、手摇计算器等原始计算工具一样，目前的各种计算辅助数学工具（例如 Mathematica）的主要用户并不是数学家，而是物理学家和工程师。更广义地说，整个人工智能领域都有同样的问题：子系统和子领域之间没什么联系，鲜有整合。

必然性（necessity）和可能性（possibility）是相对的。底层的必要性约束了上层的可能性。物理定律的可能性被更底层的数学定理的必然性所约束；依次，数学定理的可能性又被逻辑的命题和定义所约束。上层的思想（thought）恰是底层的世界（world），一类人的思想是另一类人的世界。

数学是人类智力的最高端。不明就里的哲学家对数学说三道四是令人最不能忍受的。曾有哲学家怀疑数学真理的必然性。逻辑学家霍奇（Wilfrid Hodges）曾撰文"数学中的必然性"（Hodges 2007）对这些非逻辑学家出身的哲学家表示不满。他统计了著名的代数教科书（伯克霍夫和马克雷恩的《现代代数研究》）的头一百页中出现的所有模态词（allow, can, cannot, could, essential, have to, impossible, inevitably, may, might, must, necessarily, need not, need only, possibility, possible, will），一共 340 次，平均每页 3.4 次。但这些模态词都是修辞，而不是语义。

逻辑要想有用，必须得走定理证明的路子，当所有的数学家都离不

开定理证明器时，逻辑学家才敢对趴在背上的数学家和数学家背上的那谁以及更上的那谁理直气壮。定理证明比下棋可难多了。

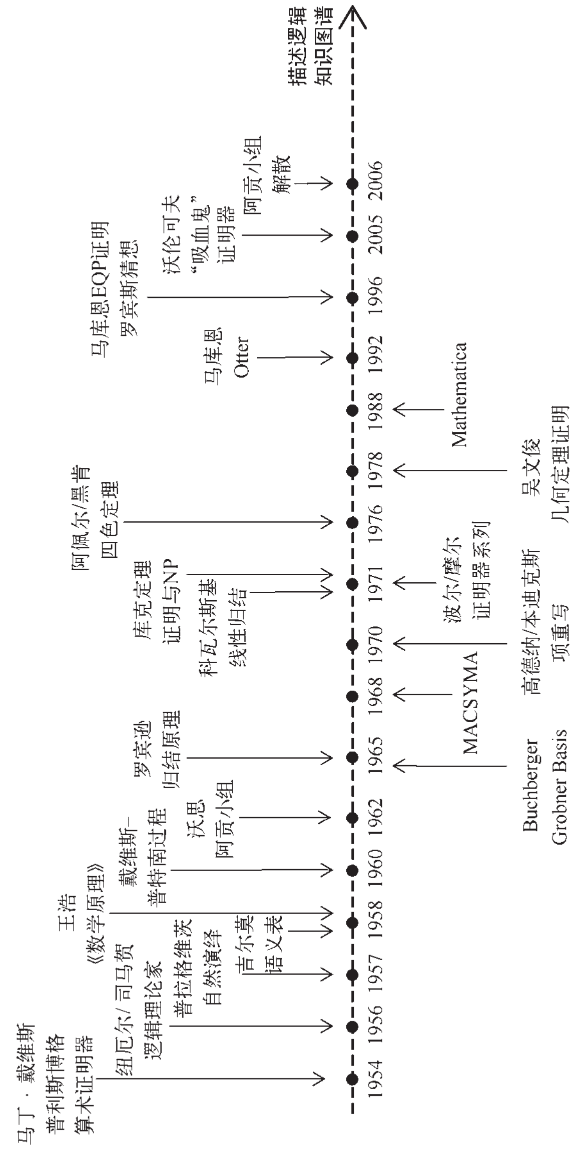
参考文献指南

除了几本教科书和那本手册，Siekmann 和 Wrightson 编辑的经典论文集是不错的参考文献。如果懂一点逻辑，想进一步探索，可参考王浩的《数理逻辑通俗讲话》，虽然是 1981 年出版的，但对于深度科普而言，不算过时。其中提到机器定理证明和计算理论。Davis (2001)和 Bibel (2007)是自动定理证明早期历史的第一手材料。20 世纪 80 年代末期出版的《人工智能手册》中有两章关于机器定理证明的内容，但取材莫名其妙，想想手册的编者是费根鲍姆一干人等，也就不会太过惊奇。

如果觉得只看文献不过瘾，真起了兴，还想动动手，不妨看看 Riazanov 的博士论文。Riazanov 是 Voronkov 的学生，师徒俩合作了高效的“吸血鬼”定理证明器。对实战者，Term Indexing 不可不懂，Graf (1996)是这方面有益的参考书。

吴文俊的生平见吴文俊《走自己的路》。王浩的生平见尼克《哲学评书》和政协齐河县文史资料委员会编《齐河文史资料第六辑》。

附录 1：自动定理证明大事记



附录 2：埃尔布朗奖

自动定理证明领域的大奖以埃尔布朗（Herbrand，或译厄布朗，国内也有按英文发音译为赫尔布兰德）命名。埃尔布朗是天才的法国数学家，他的博士论文为数理逻辑的证明论和递归论奠定了基础，当哥德尔不完全定理刚出来时，他检查了自己的论文，写了一句话作为附言：哥德尔的结果和我的结果并不矛盾。他给比他年长两岁的哥德尔写信讨论递归函数，哥德尔 1931 年 7 月 25 日给他回了信，两天后埃尔布朗爬阿尔卑斯山时因事故身亡，年仅 23 岁。一阶逻辑证明中的埃尔布朗域（Herbrand Universe）以他命名。沃思是当之无愧的第一届得奖人，吴文俊是第四届，马库恩是第七届。2015 届是"吸血鬼"的作者、曼彻斯特大学教授沃伦可夫。2016 届是以时态逻辑做程序证明的斯坦福大学退休教授马纳和斯坦福研究所的瓦丁格。

年度	获奖人	获奖原因（我的理解，没有引获奖辞）
1992	Larry Wos 拉里·沃思	除了原创性的技术贡献（例如，SoS、paramodulation、demodulation 等）外，沃思是阿贡实验室定理证明小组的创建者和领导人。这个小组为自动定理证明做出了卓越贡献
1994	Woody Bledsoe 伍迪·布莱索	布莱索出身贫寒，后从军，24 岁才上大学。有意思的是，他职业生涯早期做的是模式识别，后来坐镇得克萨斯大学奥斯汀分校后才涉足定理证明，招收了波尔和摩尔加入。他后来促成了 MCC 的成立，英曼将军任命他担任 MCC 负责 AI 的副总，他旋即招收雷纳特开始 Cyc 项目
1996	John Alan Robinson 阿兰·罗宾逊	他创建的归结原理包括一系列方法：归结、合一、包含。对整个定理证明及相关学科产生了深远影响
1997	Wu Wen-Tsun 吴文俊	几何定理证明
1998	Gérard Huet 杰拉德·厄埃	将合一算法推广到高阶逻辑，建立合一算法的理论基础。用简单类型论（typed lambda）实现高阶逻辑的定理证明

（续）

年度	获奖人	获奖原因（我的理解，没有引获奖辞）
1999	Robert S. Boyer、 J Strother Moore 波尔和摩尔	两人职业生涯的大部分是在奥斯丁分校合作波尔–摩尔定理证明器。他们将定理证明技术应用到硬件核实，尤其是 Intel Pentium 出现浮点运算错误后，AMD 的 CPU 首先使用了他们的证明器验证正确性。他们和另一位同事考夫曼（Matt Kaufman）2005 年一起赢得 ACM 软件系统大奖
2000	William McCune 威廉・马库恩	对定理证明的理论和实践都做出杰出贡献。Otter、Prover9、EQP 以及罗宾斯猜想的证明
2001	Donald W. Loveland 唐纳德・爱之地	他是 DPLL 中的第二个 L。一直坐镇杜克大学
2002	Mark E. Stickel 马克・史蒂克尔	他的贡献包括理论归结、AC 合一、PTTP（Prolog 定理证明器）
2003	Peter B. Andrews 彼得・安德鲁斯	用类型论做高阶逻辑的机器证明
2004	Harald Ganzinger 哈罗德・甘金格	理论：superposition。实践：SPASS 证明器
2005	Martin Davis 马丁・戴维斯	第一个机器定理证明程序。DP、DPLL。这个奖对戴维斯来说太迟了
2006	Wolfgang Bibel 沃尔夫刚・白贝尔	基于连接图的证明方法
2007	Alan Bundy 阿兰・邦迪	证明规划、元推理
2008	Edmund M. Clarke 埃德蒙・克拉克	因为模型检验得了图灵奖，埃尔布朗奖是稍带的。早期指导过学生赵旭东用 Mathematica 实现过一个分析定理证明器
2009	Deepak Kapur 迪帕克・卡普尔	项重写、符号计算。曾帮马库恩找到本校的教职
2010	David Plaisted 大卫・普雷斯泰	项重写
2011	Nachum Dershowitz 拿春・德施维茨	项重写的理论
2012	Melvin Fitting 马尔文・菲亭	模态逻辑的机器证明
2013	Greg Nelson 格里格・内尔逊	程序验证

（续）

年度	获奖人	获奖原因（我的理解，没有引获奖词）
2014	Robert L. Constable 罗伯特・康斯特波	程序正确性证明
2015	Andrei Voronkov 安德烈・沃伦可夫	高效定理证明器的实现者，代表作“吸血鬼”
2016	Zohar Manna 佐哈・马纳 Richard Waldinger 理查・瓦丁格	两人一直合作用时态逻辑证明程序的正确性。有多本专著

附录 3：几本自动定理证明教科书的评论

本书除了梳理历史，还想有科普的功能。理论上说，如果有一本好的教科书，就不需要依赖原始文献了——想了解牛顿第二定律不一定非得读牛顿本人写的《自然哲学的数学原理》。但自动定理证明学科恰缺乏合适的教科书。早期最有影响的教科书当属 Chang and Lee (1973)。本书两位作者 20 世纪 60 年代末在加州大学伯克利分校读博士时是同学，那时正是定理证明发展的高峰期。作者之一 Lee 就是李家同（Richad Char-tung Lee），他的曾祖父李瀚章是李鸿章的长兄。李家同后来回到中国台湾做了“清华大学”代理校长和暨南大学校长，也是成功的散文作家。这本书的整体风格非常好，而且有大量习题，无论教学还是自学皆可。但书中第 7 章存在几个致命的错误，否则真可以成为本领域的经典。本书的另一作者 Chang 则一直待在美国，长期在 IBM 工作，后来又写了一本人工智能的教科书，但再无大名。两位作者的不同境遇可作为留学生选择海归还是留美的参照。后来两位都远离定理证明领域，该书也没再出修订版，很可惜。

Loveland (1978)，风格类似 Chang and Lee (1973)，但更严格，但内容业已老化。

Wos (1983)是一本很容易读的书，但很少讲逻辑和定理证明技术的原理，更像是阿贡实验室早期定理证明器的使用手册。阿贡系作者的所有书都有类似的特点。

Bibel (1987)虽是教科书体例，但内容偏重作者自己的连接方法，目前已不是主流。

Burris (1997)主要是逻辑的教科书，但书中专有一章讲方程逻辑，详细地讲解了 Knuth-Bendix 过程，是方程逻辑和项重写很好的入门材料。

Nerode and Shore (1997)的两位作者都是逻辑学家。整体内容严谨。缺憾是比较偏重 Prolog，定理证明的技术只讲了线性归结。

Gallier (2003)的体例很像 Nerode and Shore (1997)，讲过归结后讲了 SLD 归结，这是 Prolog 的理论基础。这两本书我更喜欢 Nerode and Shore (1997)。

Fitting (1995)也是一本入门书，讲了归结，也讲了表方法（Tableaux）。作者是埃尔布朗奖得主。

Robinson and Voronkov (2001)是手册型。两位编者一位是归结原理的发明者，另一位是目前最厉害的定理证明器“吸血鬼”的主要作者。内容现代、完整。但不是教科书形式，初学者读起来会很费劲。

不过话说回来，如果现在非要我推荐一个定理证明教科书的作者，Voronkov 应该是第一人选。好的教科书的缺乏也反映了这个学科的凋零。至于进一步的学习，不严肃的读者看完本篇文章就可以打住，看多了耽误您前程。严肃的读者可先从 Nerode and Shore (1997)下手，然后看看 Burris (1997)方程逻辑那一章，对个别议题感兴趣的再去看 Robinson and Voronkov 的《自动推理手册》（*Handbook of Automated Reasoning*）。

第 3 章
从专家系统到知识图谱

The test of all knowledge is experiment.
实验是知识的试金石。
——Feynman Lectures on Physics（《费曼物理学讲义》）

1. 费根鲍姆和 DENDRAL

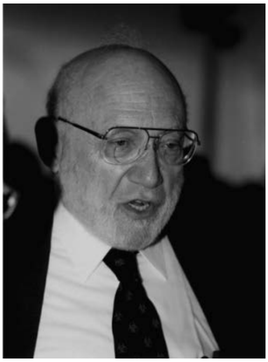
费根鲍姆进入卡内基理工学院（卡内基梅隆的前身）攻读电气工程（简称 EE）本科时才 16 岁。大三时一门“社会科学的数学模型”的课设定了他的人生轨迹，那门课的老师是司马贺。本科毕业后他留校，在司马贺任院长的工业管理研究生院读博士。博士毕业后他来到加州大学伯克利分校的工商管理学院任教。他曾和师弟菲尔德曼（Julian Feldman）合编过一本论文集《计算机与思维》，这本文集的版税后来被用来在国际人工智能联合会（IJCAI）资助“计算机与思维”奖，这成为人工智能界给 40 岁以下青年学者最重要的奖项，初衷有点像数学界的菲尔茨奖。排在一长串获奖人第一位的是维诺格拉德（Terry Winograd），其中还有雷纳特（Douglas Lenat）、英年早逝的马尔以及吴恩达，最新的一位（2016 年度）是斯坦福大学做自然语言处理的新秀 Percy Liang。1962 年麦卡锡从东岸的麻省理工学院搬到美丽的旧金山湾区，组建了斯坦福大学计算机系。1964 年费根鲍

姆响应麦卡锡的召唤，离开伯克利，到不远处的斯坦福大学协助麦卡锡。


费根鲍姆（1936—　）

1958 年李德伯格（Joshua Lederberger）获得诺贝尔生理奖时才 33 岁，得奖的第二年，他就离开当时任教的威斯康星大学前往加州，受邀重建斯坦福大学的医学院并担任遗传学系主任。那时，斯坦福大学的医学院还在旧金山，和公立的加州大学旧金山分校一起。加州大学其他各分校没有医学院，旧金山分校就是整个加州大学的医学院，直到 20 世纪 90 年代末期，斯坦福医学院和加州大学旧金山分校分久必合，寻求合并但最终未果。话说回来，李德伯格在哥伦比亚大学读本科时就受到“莱布尼茨之梦”的影响，企图寻找人类知识的普遍规则。1962 年夏，李德伯格还在斯坦福计算中心听编程的课，他上手的第一门语言是 BALGOL[①]。他很快就结识了刚从麻省理工学院加入斯坦福大学的麦卡锡，他们还企图把明斯基也吸引到斯坦福医学院。

① 宝莱公司（Burroughs）版的 Algol。



李德伯格（1925—2008）

费根鲍姆 1964 年在斯坦福大学高等行为科学研究中心的一次会上见到了李德伯格，对科学哲学的共同爱好促成了他们漫长而富有成效的合作。那时李德伯格的研究方向是太空生命探测，更具体地说就是用质谱仪分析火星上采集来的数据，看火星上有无可能存在生命。费根鲍姆的兴趣则是机器归纳法，用现在的话说就是机器学习。他们俩，一个有数据，一个搞工具，一拍即合。从科学史的角度看，这是个跨学科的合作，李德伯格的影响力和领导力起了核心作用。按照布坎南的说法，以费根鲍姆为首的计算机团队的任务就是把李德伯格的思路算法化。李德伯格完成哲学构思后就兴趣迁移了，他最初的想法花了费根鲍姆们 5 年的时间才得以实现，李德伯格责怪他们太慢了。

费根鲍姆很快就发现李德伯格是遗传学家，对化学其实也是一窍不通，于是他们找到同校的化学家兼作家兼口服避孕药发明人翟若适（Carl Djerassi）帮忙。翟若适没得过诺贝尔奖，但他得过美国国家科学奖（得奖人包括维纳、哥德尔、香农、丘成桐）和美国国家技术与创新奖（得奖人包括杜邦公司和 HP 创始人帕卡德、Intel 创始人诺伊斯、微软创始人盖茨），这是非常独特的。另一位两个奖都得过的是发明了计算机 RISC 架构的寇克（John Cocke）。翟若适那时刚从韦恩州立大学转到自由的斯坦福大学，

李德伯格是他在加州结识的第一个朋友。三人合作的结果就是第一个专家系统 DENDRAL。DENDRAL 输入的是质谱仪的数据，输出是给定物质的化学结构。费根鲍姆和他的学生捕捉翟若适和他的学生的化学分析知识，把知识提炼成规则。这个专家系统有时做得比翟若适的学生还准。在翟若适的大部头自传中，只有一小段提到 DENDRAL，这个项目在他成果辉煌的学术生涯和多姿多彩的生活中，实在算不上什么。翟自传中说费根鲍姆一直把 DENDRAL 的核心称为“翟算法”，而布坎南则记得大家都认为专业知识的提供者是李德伯格，也许是费根鲍姆圆滑，也许是计算机团队更多地接触李德伯格。

费根鲍姆是学术活动家，刚到斯坦福大学就担任计算中心主任，当时这个职位比计算机系主任恐怕还更有影响力。20 世纪 60 年代初期和中期，费根鲍姆曾两次访问苏联，对苏联的计算机科学和控制论研究印象深刻，他早就观察到苏联的研究偏理论而轻实践。但苏联下棋程序的胜利确实让世界吃惊。苏联控制论的定义太宽泛，无所不包，其结果也没有重点，难有突破性成果，而当时中国的自动化学科就是仿效苏联的。美国不存在自动化学科，无所不包的 EE 和自动化有很大交集。费根鲍姆意识到他的苏联同行企图利用他的名望来为苏联同行背书并争取资助。而在美国，动态规划的发明人贝尔曼则通过军方智库兰德公司给空军建议美国应该提防苏联的计算机科学研究。费根鲍姆不爽贝尔曼的报告，认为这是他利用苏联威胁论来为自己争取科研红利。而若干年后，费根鲍姆却利用日本的五代机项目宣扬日本威胁论，这不得不让人怀疑他的动机。费根鲍姆创办的数家公司因各种原因，都没有大的成功。其中 Teknowledge 公司没成，倒是副产品、知识库项目 SUMO 留了下来，现在开源了，成为几大基础的常识知识图谱之一。

2. MYCIN

MYCIN 的牵头人布坎南也是 DENDRAL 的核心成员。布坎南是哲学出身，兴趣广泛。1964 年，在密执安州立大学学哲学的布坎南想在系统开发公司（SDC）找份夏季实习的工作，没曾想 SDC 居然把他的简历发给了兰德公司，敢情国防口的简历也共享。当时在兰德做夏季工作的费根鲍姆给布坎南打了个电话，于是布坎南到兰德公司实习并和费根鲍姆结缘。布坎南的研究方向是科学发现，他走的是逻辑路数而不是心理路数，没曾想费根鲍姆也对科学哲学兴趣浓厚。事实上，费根鲍姆和李德伯格最早关于 DENDRAL 的文章中提到了“机械化科学推理”（mechanizing scientific inference）的概念。布坎南博士毕业后想去教哲学，请费根鲍姆写推荐信，但费说服布坎南到斯坦福大学和他一起搞真正的科学发现。布坎南的哲学背景帮助了他，在 DENDRAL 项目的开头，李德伯格和费根鲍姆都没想到假设生成和理论生成的区别，同时布坎南也意识到他在哲学课堂里学到的卡尔纳普理论在计算上行不通。整个 DENDRAL 团队中没有一个人对其中涉及的化学知识有全部的了解，每个人都假设其他人知道。布坎南早期的演讲开头都得讲点化学背景知识，听众听不懂也不耐烦，他记得有一次是麦卡锡站出来对听众大喊：“你们就不能好好听着吗？”（Just listen，will you？）麦卡锡的威望为他解了围。

DENDRAL 获得成功后，布坎南开始寻找新的方向。实验科学与理论科学比较，是相对原始的，原始经验也相对容易转换成规则。除了化学和生物学，医学是另一个可以马上利用专家系统的领域。此时的斯坦福医学院来了一位从哈佛大学本科数学毕业的高才生肖特莱福（Edward Shortliffe），他 1976 年在斯坦福医学院拿到医学学位 M.D.，但一年前在布坎南的指导

下，他已经获得了计算机科学的博士，论文就是专家系统 MYCIN，一个针对细菌感染的诊断系统。MYCIN 的处方准确率是 69%，当时专科医生的准确率是 80%，但 MYCIN 的成绩已经优于非本专业的医生。肖特莱福因此获得 1976 年 ACM 为青年计算机科学家设立的霍普（Grace Murray Hopper）奖。肖特莱福随后去麻省总医院做了三年内科住院医生，1979 年回到斯坦福大学兼任医学院和计算机系的教授。

MYCIN 团队认为 DENDRAL 是专家系统的始祖，这一方面考虑了 DENDRAL 的时间点确实更早，另一方面布坎南本身就出自 DENDRAL。但纽厄尔，作为外人，却认为 MYCIN 才是专家系统的鼻祖，因为 MYCIN 首创了后来作为专家系统要素的产生式规则：不精确推理。DENDRAL 的初衷则是从专家采集来的数据做机器归纳，或者说机器学习。

虽然 MYCIN 从来没被临床使用过，但 MYCIN 的开发原理后来逐步被提炼成为专家系统核心 EMYCIN 的基础。EMYCIN 的动机是两方面的，除了通用化外，政府资助也是一个原因。20 世纪 70 年代初，DARPA 削减对人工智能的资助，原来的长期资助改为每年评审。每次向 DARPA 汇报时，费根鲍姆团队都得事先对词，他们不敢说研究经费被用来做医学相关的研究。直到后来他们拿到美国国家健康研究院（NIH）和美国国家医学图书馆（NLM）的资助，情况才有所好转。

3. 专家系统的成熟

一个领域成熟与否的主要测度之一是赚不赚钱。人工智能没有商业应用一直是被批评的原因之一。专家系统时代最成功的案例是 DEC 的专家配置系统 XCON。DEC 是 PC 时代来临之前的宠儿，他们用小型机冲击 IBM。

当客户订购 DEC 的 VAX 系列计算机时，XCON 可以按照需求自动配置零部件。从 1980 年投入使用到 1986 年，XCON 一共处理了八万个订单。

XCON 到底为 DEC 省了多少钱一直是个谜，最高的说法是一年省四千万美元，还有的说法是两千五百万，最低的说法是顶多几百万。无论如何，DEC 是把 XCON 当作商业成功来宣传的。XCON 确实体现了技术的进步，其起源是卡内基梅隆的 R1。说来也有意思，最早的 XCON 居然是用 Fortran 语言写的，失败后居然令人发指地改用 BASIC 语言。纽厄尔的博士生 Charles Forgy 发明了 Rete 算法和 OPS 语言，极大地提高了专家系统的效率，XCON 迅速采用 OPS 和随后的 OPS5。

20 世纪 80 年代初到 20 世纪 90 年代初，专家系统经历了十年的黄金期，随着日本五代机的幻灭，“专家系统”变成了一个不仅不时髦，反而有负面含义的词。互联网催生的电子商务，有很多和 XCON 类似的应用场景，于是新瓶装旧酒，专家系统摇身一变，改名规则引擎，成为中间件的标配。征信、反欺诈和风险控制一直是规则系统擅长的领域，征信公司 FICO 收购了一系列一直苟延残喘的专家系统公司，包括 Forgy 的 RulesPower。目前已经很少有独立的专家系统公司了。

4. 知识表示

知识表示一直是人工智能不温不火的一个领域，催生者是专家系统和自然语言理解。KRL（Knowledge Representation Language）是最早的知识表示语言之一，有影响但不成功。参加过施乐实验室（XEROXParc）KRL 项目的维诺格拉德在多年后总结教训时说，KRL 要同时解决两个问题：第一，知识工程师的可用性，也就是说人可读可写；第二，得有底层的麦卡

锡风格的逻辑来支撑语义。要同时解决这两个互相矛盾的问题，必然导致结果太复杂，四不像，知识工程师和逻辑学家都不买账。

逻辑

逻辑是最方便的知识表示语言，从亚里士多德开始人们就熟悉，逻辑同时具有各种数学性质。任何一本逻辑入门书都会有那个著名的苏格拉底的例子：人必有一死，苏格拉底是人，所以苏格拉底必死。这个三段论表示成现代的数理逻辑就成了如下形式。

大前提和小前提：$(\forall x)$ Man$(x) \supset$ Mortal(x) & Man(Socrates)

结论：Mortal(Socrates)

一阶逻辑也称谓词逻辑，是希尔伯特简化罗素的《数学原理》中逻辑的结果。谓词逻辑没有本体，也就是没有关于特定世界的公理。也正因此，哲学家、逻辑学家蒯因把逻辑等同于一阶逻辑。一阶逻辑只是语法，没有本体，没有语义；而高阶逻辑，在蒯因的眼里，其实是“披着伪装的集合论”（set theory in disguise）。费根鲍姆所谓的知识就是本体。当然，费根鲍姆不是从逻辑的角度而是从心理的角度看问题，这显然受到他的老师纽厄尔和司马贺的影响。

可计算性和计算复杂性理论与逻辑密不可分。一阶逻辑是不可判定的，命题逻辑的可满足性问题是 NP 完全的。知识表示的一个核心问题是找到一个一阶逻辑的子集，它是可判定的，并且尽可能地有效。描述逻辑应运而生。描述逻辑可以表达实体和类以及类和类之间的关系。描述逻辑中的实体就是一阶逻辑中的常量。实体的表示在描述逻辑中也称 Abox，例如“牛顿是物理学家”可以表示为：

$$\text{Physicist(Newton)}$$

描述逻辑中不需要变量，描述逻辑的术语更像是集合论，类和类之间的关系也称 TBox。例如，在一个本体中，律师事务所（Lawfirm）是公司（Company）的子集，公司是组织（Organization）的子集，组织是 Agent 的子集，Agent 是 Thing 的子集，这样一系列关系可以表示为：

律师事务所 ⊑ 公司 ⊑ 组织 ⊑ Agent ⊑ Thing

其对应的一阶逻辑表达式为：

$$\text{Lawfirm}(x) \rightarrow \text{Company}(x), \text{Company}(x) \rightarrow \text{Agent}(x),$$

$$\text{Agent}(x) \rightarrow \text{Thing}(x)$$

一阶逻辑的定理证明的 Term Index 技术中有 Subsumption 的概念，表示 Term 之间的集合从属关系。Tbox 表达了一种简化的 Subsumption。除了 ABox 和 TBox，还有 RBox 表示关系或者角色，关系之间可以有集合论中常有的子集、交集、并集等操作，例如“父亲的父亲是祖父”可表示为：

$\text{has Father} \circ \text{has Father} \sqsubseteq \text{has GrandFather}$

其对应的一阶逻辑表达式为：

$\text{has Father}(x, y) \wedge \text{has Father}(y, z) \rightarrow \text{has GrandFather}(x, z)$

心理学与语言学

知识表示的另一个来源是心理学和语言学，例如概念的上下位继承关系最方便的表示方式是树而不是一阶逻辑。心理学实验表明人在回答“金丝雀会飞吗？”要比回答“鸟会飞吗？”花的时间长，要回答第一个问题，人要再做一次“金丝雀是鸟”的推理。因为人在存储知识时只存储抽象的，这是空间经济的考虑。心理学家米勒和乔姆斯基等一起开拓了认

知科学，他最出名的论文大概就是那篇“魔力数字七”（The Magic Number Seven）。他除了理论的贡献，晚年带领普林斯顿大学的认知科学实验室同仁做了“词网”（WordNet）。WordNet 不单是一个同义词辞典，还定义了词的上下位关系，例如，car 的一种上位是 motor，可以再上位到 wheeled vehicle，直到 entity。WordNet 成为自然语言处理的基本工具。

```
Sense 1
car, auto, automobile, machine, motorcar
   => motor vehicle, automotive vehicle
     => self-propelled vehicle
       => wheeled vehicle
         => vehicle
           => conveyance, transport
             => instrumentality, instrumentation
               => artifact, artefact
                 => whole, unit
                   => object, physical object
                     => physical entity
                       => entity
         => container
           => instrumentality, instrumentation
             => artifact, artefact
               => whole, unit
                 => object, physical object
                   => physical entity
                     => entity
```

WordNet

明斯基的框架

框架（Frame）就是类型。金丝雀是鸟，所有鸟的性质自动流传给金丝雀，鸟能飞，金丝雀也能飞。苹果手机是手机，手机能打电话，苹果手机也能打电话。框架导致了面向对象（OO，Object-Oriented）的设计哲学，相关的程序设计语言都受此影响。从这个意义上还真验证了：当一个概念

有了成熟的实现时，就自动脱离了人工智能。

差不多同时出现的语义网络（Semantic Net，注意这个和后面讲到的 Semantic Web 相关但不同）是与框架等价的表示方式。语义网络中的每个节点就是一个框架，每个节点上的边可以看作一个 slot。

Sowa 的概念图

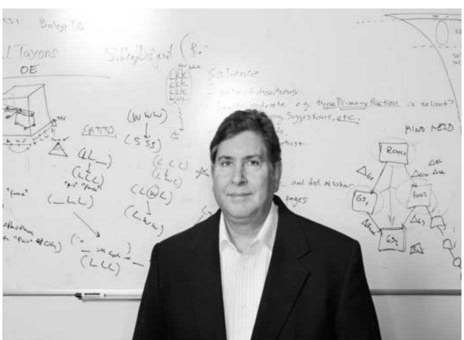
IBM 的索瓦（John Sowa）在 20 世纪 80 年代初提出“概念图”（Conceptual Graph），企图把知识表示奠定在更加坚实的数学和逻辑基础上。大约同时或稍早，德国数学家威勒（Rudolf Wille）提出基于代数的“形式化概念分析”（Formal Concept Analysis）。程序设计语言理论也变得越发严谨。在概念图中，多重继承的类型层次（Muliple Inheritance）可以用代数的偏序关系“格”（Lattice）来表示。“全序”关系（total-order）是“偏序”的一个特例。一个全序集中的成员要么是 $a<=b$，要么是 $b<=a$。偏序关系容许一个成员可以有多个上级，也可以有多个下级。而全序集中，每个成员只能有一个上级和一个下级，所以，全序关系有时也被称为线性关系。当用“格”做知识表示时，每个概念就是“格”中的一员，概念之间服从偏序关系。多重继承的 OO 都是偏序关系。

5. 雷纳特和大知识系统

在日本五代机项目带来的狂潮中，美国政府决定联合多家高技术企业在得克萨斯大学奥斯汀分校所在地建立微电子与计算机技术公司（MCC，Micro electronics and Computer Technology Corporation）以抗衡日本，海军上将英曼（Inman）被任命为 CEO，当时在奥斯汀分校从事机器定理

证明的资深教授布莱索（Woody Bledsoe）全职加入 MCC 负责研发。这让人想起二战时曼哈顿工程中格里菲斯将军和奥本海默的分工。费根鲍姆提议建立美国的国家知识技术中心（National Center for Knowledge Technology），像狄德罗创建百科全书一样，把人类有史以来的知识建库，这自然对 MCC 的计划也有很大影响。布莱索向英曼推荐了费根鲍姆的学生雷纳特（Douglas Lenat）。

雷纳特此时 30 出头，是人工智能领域的一颗新星。他在宾夕法尼亚大学得了数学和物理双学位后又拿了个数学硕士，毕业后他对数学和物理的学术工作都失去了兴趣，但他毕业马上就面临征兵，只得又跑到加州理工学院接着读博士。期间他对人工智能产生了强烈兴趣，遂转学到斯坦福大学想跟随麦卡锡，但正赶上麦卡锡的学术休假年，于是他变成了费根鲍姆和布坎南的学生。他的博士论文实现了一个称作 AM 的程序，为此，IJCAI 在他博士毕业第二年就给了他一个“计算机与思维”大奖。AM 就是自动数学家（Automated Mathematician）的简称，它可以自动“发现”定理。雷纳特没有用“发明”这个词，从某种意义上体现了他的哲学立场。在经受了一连串关于 AM 不严谨的批评之后，雷纳特推出了 AM 的后继 Eurisko。Eurisko 的应用领域更加广泛，包括博弈。



雷纳特（1950— ）

当雷纳特来到 MCC 时，他已经有了一个新的想法：把人类的常识编码，建成知识库。这个新项目叫 Cyc，这三个字母取自英文单词“百科全书”（encyclopedia）。这其实就是最早的知识图谱。雷纳特坚定地支持他老师费根鲍姆的知识原则（Knowledge Principle）：一个系统之所以能展示高级的智能理解和行为，主要是因为在所从事的领域所表现出来的特定知识：概念、事实、表示、方法、比喻以及启发。雷纳特甚至说：“智能就是一千万条规则。”

索瓦提出“知识汤”（knowledge soup）的说法：我们脑子里的知识不是一坨知识，而是好几坨知识，每一坨内部是一致的，但坨和坨之间可能不一致，坨和坨之间是松散耦合的。古哈（Guha）在斯坦福大学的博士论文导师是麦卡锡和费根鲍姆，他的论文讲的是如何将一个大理论分解为多个“微理论”（microtheory），如何利用 Cyc 作为多个不同数据源的前端而不是全部，这恰是索瓦的“知识汤”的实现。Cyc 由此可成为数据或信息整合的工具。雷纳特对此有点不爽，但他还是把古哈招到了门下。

雷纳特对 Cyc 自视甚高。他 1984 年时曾预言 15 年后，也就是 1999 年，每台马路上卖的电脑里都得预装 Cyc。1986 年，雷纳特再度预言：Cyc 如果可用的话，至少要有 25 万条规则，这至少要花 350 个人年，也就是 35 个人干十年。Cyc 项目开始时有 30 个左右的知识工程师，他们每天的工作就是利用 Cyc 的语言 CycL 把日常生活的常识编码，这包括教育、购物、娱乐、体育等。到了 1995 年，日本的五代机项目烟消云散，美国政府也削减了对 MCC 的支持。雷纳特带着 Cyc 离开 MCC，成立 Cycorp 公司，开始了漫长的创业路程。核心骨干古哈离开 MCC，先后加入了苹果、网景和谷歌三家公司。

倒是 WordNet 在各种版本的 Linux 配置的 App Center 里很容易找到。

WordNet 比 Cyc 更基本也更好用，当然 WordNet 没有 Cyc 那么多的推理功能。再过 50 年，人们对一阶逻辑也不会像对莎士比亚那么熟。也许 WordNet 并不是一个好的例子。Cyc 的原始目标更像是当今的维基百科，不过维基百科的受众是人，Cyc 的用户是机器。Cyc 在 20 世纪 90 年代初期就被批评说没有成功案例，而当时的其他专家系统都有或多或少的应用。雷纳特辩解道，Cyc 只有在知识量突破临界点（critical mass）之后才能带来收益。现在离开那时的批评，已经又过去了 20 多年。我们还是看不到可观的应用。

Cyc 现在有两个版本：企业版和研究版。企业版收费，研究版对研究人员开放。曾经有一个开源的 OpenCyc，是一个简版，但试用中发现 OpenCyc 引发的问题太多，被停掉了，Cyc 正在准备用一个云版代替 OpenCyc。

雷纳特曾说："学习只在已知事物的边缘发生，所以人们只可能学到与自己已知相似的新东西。如果你试图学习的东西与你已知的东西距离不远，那么你就能学会。这个边缘的范围越大（你已知的东西越多），就越有可能发现新的东西。"这不仅是他早期研究机器学习的感悟，也可以看作他对后来 Cyc 项目的体会。1984 年雷纳特开始 Cyc 项目时，才 30 岁出头，现在 30 多年过去了，他已经年近 70 岁，仍然担任 Cycorp 的 CEO。

6. 语义网

由专家系统一脉相传的这一派自身的逻辑功力不够，他们一直在和定理证明派掐架；另一方面，他们的工程实践又略显欠缺。专家系统风过了后，他们变成了暗流，直到歪打正着的万维网支持者之一伯纳斯–李（Tim Berners-Lee）提出"语义网"（Semantic Web，见 Berners-Lee 2001），他们认为机会来了。伯纳斯–李因为草根且便捷的 HTTP 协议和超文本链接

标准 HTML 出了名，被各种媒体称为万维网的发明人。第一波互联网热之后，他马上离开欧洲粒子中心，到麻省理工学院新创办的万维网协会（W3C）担任理事长。麻省理工学院给他在当时的计算机科学实验室（现已合并为 CSAIL 计算机科学与人工智能实验室）谋了个位置，显然目的是提高学院在互联网大潮中的影响力。互联网热拉大了美国科技创新之都硅谷和麻省理工学院所在波士顿 128 公路之间的距离。20 年后，伯纳斯–李不负所望，得了 2016 年图灵奖，这大概是图灵奖有史以来含金量最低的一个。

其实万维网更大的功劳应该算在天才程序员安德森（Marc Andreessen）的头上，是革命性的 Mosaic 浏览器带来了互联网革命。年轻的安德森志在改变世界，而不是徒得虚名。在克拉克（Jim Clark）的指点和帮助下创办了互联网标志性企业“网景”后，他又经历了几次艰难但不是特别成功的创业。在第二次互联网高峰来临时，安德森与时俱进地创办了新一代风险投资公司安德森–霍洛维茨[①]，成果和影响力直追老牌风投 KPCB 和红杉资本。

话说回来，得益于 20 世纪 80 年代就已成熟的 SGML 标准，超文本链接标准 HTML 是 SGML 的某种不够深谋远虑的简化版。而 HTTP 顶多算挂在浏览器伟岸身躯上的一个可有可无的小玩具，直到互联网标准化组织 IETF 对 HTTP 做了几次修改之后，HTTP 才更像是个专业的东西。万维网协会（W3C）的目的是为万维网设立标准。伯纳斯–李身边一下子聚集了一帮多年不得志的非主流 IT 从业者。他们在 W3C 中提出的几个乱七八糟的标准确实体现出他们的理论功底之缺乏。在 W3C 的各种会中经常会见

① 网站域名 a16z.com 用了安德森的第一个字母和霍洛维茨的最后一个字母，以及两个姓的长度。

到各大技术公司中游离于边缘的资深从业者，有些人换了工作还是代表不同的公司参加各种标准化组织的工作组会议，他们的职业生涯不是为了做出技术贡献，而是不断为自己的存在找各种高尚的理由并脱离公司的管理体制。2006 年美国人工智能年会（AAAI）上，在伯纳斯-李的主题发言之后，时任谷歌研发总监的诺维格（Peter Norvig）尖锐地发问，被人认为是毫不留情地批评了语义网。

W3C 语义网的工作后来在一些准逻辑学家加入后引入了描述逻辑，变得貌似严格起来，经过几次迭代后演化成大杂烩，理论上不严谨，实践中不可用。所谓“万事开头难”，但开了一个坏头，则是灾难，为后人修正制造了人为障碍。我们可以把语义网的工作与早期的 DENDRAL 和 MYCIN 做个对比，很明显，无论理论、实践还是人文社会政治环境，都不可同日而语。几乎每个“语义网”的项目都能看到古哈的影子，2013 年他还在谷歌时曾有个演讲“隧道深处见到光”（Light at the End of the Tunnel），与其说是夸耀成功，倒不如说是总结教训。

7. 谷歌和知识图谱

在维基百科的同时，还有个 Freebase。维基百科的受众是人，而 Freebase 则强调机器可读。2016 年维基百科达到 1000 万篇文章，其中英文版达到 500 万篇文章，而 Freebase 有 4000 万个实体表示。Freebase 的背后是一家名叫 Metaweb 的创业公司，创始人之一是希利斯（Danny Hillis）。2010 年 Metaweb 被谷歌收购，谷歌给它起了个响亮的名字“知识图谱”。2016 年谷歌对 Freebase 停止更新，把所有数据捐给维基数据（Wikidata）。维基数据是维基百科的母公司 Wikimedia 的德国分部的项目，得到了微软创始人

艾伦出资创办的艾伦人工智能研究所的支持。

除了维基数据之外，另外还有几个开源的知识图谱，如 DBpedia、Yago、SUMO 等。值得一提的是，SUMO 是费根鲍姆创办的一家失败的公司 Teknowledge 的遗产。所有开源的知识图谱的基础数据的重要来源之一都是维基百科。以维基百科中居里夫人的词条为例，在“居里夫人”页面的右边，有个被称为 infobox 的框，里面包含了居里夫人的数据，例如她的生日、卒日、出生地、母校、老师、学生等，这些数据已经接近结构化的质量。



维基百科中“居里夫人”词条

IBM 沃森的底层就整合了两个开源知识图谱 Yago 和 DBpedia。在常识图谱的上面还可以构建垂直领域（例如生物、健康、金融、电商、交通等）的专业图谱。

纽厄尔和司马贺在人工智能中是符号派。其实，符号派中也派中有派，比司马贺一支更加“符号”的是机器定理证明，纽厄尔和司马贺的早期生涯曾和一帮逻辑学家结下梁子，而费根鲍姆继承了老师的基因，对定

理证明的第二代代表人物阿兰·罗宾逊极尽攻击之能事。明尼苏达大学巴贝奇研究所做口述历史的诺伯格在采访各位人工智能前辈时，总想把符号派归结到麻省理工学院和卡内基梅隆大学之争，而在斯坦福大学的麦卡锡和 SRI 的尼尔森偏麻省理工学院，同在斯坦福大学的费根鲍姆则偏自己的母校卡内基梅隆大学。当然我们还可以溯源到更早的达特茅斯会议上麦卡锡和司马贺结下的梁子。但归根结底，专家系统的理论基础依然是机器定理证明。尽管费根鲍姆从某种意义上人造了“知识与推理”的对立话题，并强调知识对于逻辑推理的重要性，但知识和推理是不可分割的一对，强调知识并不能让你脱离符号派。如果从纯粹的定理证明的角度简单地看专家系统，所谓知识其实就是公理，公理越多，推理的步骤自然就会越少。所谓知识和推理的对立，其实是狭义（特殊目的）和广义（通用）的区别。知识是狭义的，推理是广义的，因为不需要过多的公理。狭义对机器的短期实现高效，但人的学习门槛较高；而广义对机器的实现自然低效，但人学习的门槛较低。一阶逻辑的学习门槛最低，但当知识库变大，推理引擎也得变得更加专用才能高效。

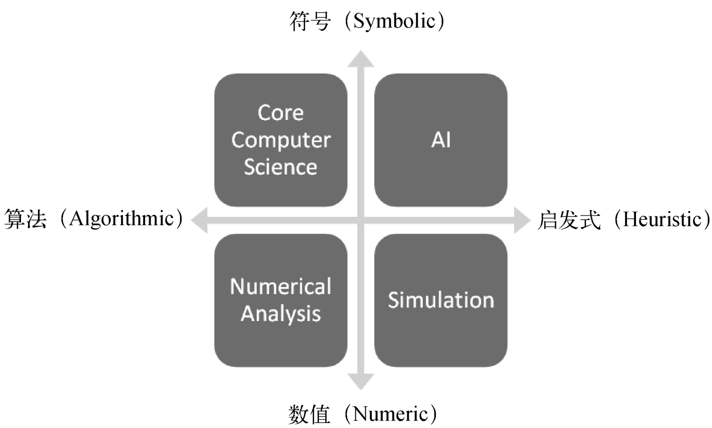
参考文献指南

DENDRAL 项目的来龙去脉见 Lindsay，Buchanan，Feigenbaum，Lederberg (1993)。李德伯格 1987 年的文章值得一读，除了关于 DENDRAL 的人和事的回忆，还有他对科学方法的体会和总结。MYCIN 项目见 Buchanan (2006) 的文章和 Buchanan , Shortliffe (1984)的书。纽厄尔在给后者写的前言中指出 MYCIN 才是真正的第一个专家系统。

布坎南在 1991 年曾接受明尼苏达大学巴贝奇研究所的口述历史采

访，见 Buchanan (1991)。在布坎南的采访中，我们清晰地看到一些学术管理者在团队中举足轻重的作用，例如斯坦福大学人工智能实验室的副主任（Executive Officer，美国海军中简称 XO，特指副舰长）厄内斯特（Les Earnest），他为麦卡锡和费根鲍姆打理人工智能实验室的日常事务，撰写给各种资助机构的项目计划书，而在麻省理工学院为明斯基操持家务的是温斯顿（Patrick Winston）。

布坎南在回忆 1965 年到 1985 年主要的人工智能事件时，提到两件事，一个是硬件进步（LISP 工作站是代表例子），另一个是基于知识的系统。布坎南把计算机科学分成两个维度，横轴是处理，纵轴是数据，这多少有点像知识和推理之分。数据可按类型分成数值的和符号的，而处理则分为算法的和启发式的，那么整个计算机科学可以划分成如下四个象限：



计算机科学的划分

Cyc 项目见 Lenat and Guha (1989)。描述逻辑从浅到深有各种文献，Rudolph (2011)是一篇中间适度的文章。

第 4 章
第五代计算机的教训

People learn from history that people never learn from history.
历史的教训就是历史从来没给过人教训。
——Georg Wilhelm Friedrich Hegel（黑格尔）

Those that fail to learn from history, are doomed to repeat it.
不懂吸取教训的人注定会重蹈覆辙。
——Winston Churchill（丘吉尔）

1. 背景

1978 年，日本通产省（MITI，Ministry of International Trade and Industry）委托日本计算机界大佬、时任东京大学计算机中心主任的元冈达（Tohru Moto-Oka）研究下一代计算机系统。当时的计算机工业按照电路工艺划分计算机的发展：第一代计算机是电子管，第二代是晶体管，第三代是集成电路，第四代是超大规模集成电路（VLSI）。通产省决定三年后开始建造第五代计算机，这是日本雄心勃勃的从制造大国到经济强国转型计划的一部分，用今天的话说，他们认为日本已经进入“无人区”，日本必须搞“双创”。首创第五代计算机，可以建立日本在全球信息产业的领导地位。


元冈达（1929—1985）

三年后的 1981 年，以元冈达为首的委员会认真提交了一份长达 89 页的报告，他们认为第五代计算机不应再以硬件工艺为划分，更应看重体系结构和软件。这份报告的题目就是《知识信息处理系统的挑战：第五代计算机系统初步报告》。报告提出了 6 种先进的体系结构：(1) 逻辑程序机，(2) 函数机，(3) 关系代数机，(4) 抽象数据类型机，(5) 数据流机，(6) 冯诺伊曼机上的创新。元冈达 20 世纪 70 年代在美国做过访问学者，对数据流机和数据库机比较熟悉。和后来人们的印象不同，这份报告还是兼收并蓄的，更像是一份对当时各种技术的综述，并不是一上来就一边倒地搞逻辑程序。

其实这几种体系架构当时早就有欧美的大学在做研究，甚至已经有创业公司，例如专攻函数程序语言 LISP 硬件的 LMI 和 Symbolics 都是从麻省理工学院分出来的创业公司，在 1981 年就已经开张。话说 Symbolics.com 还是第一个.com 域名。报告中提出的函数机的性能目标是比在通用机上的软件实现快两到三倍，恐怕 Symbolics 当时就已经做到了。用硬件实现关系数据库和面向客体（Object-Oriented）的系统也不新鲜。有意思的是第 6 项，创新型冯诺伊曼机的需求：VLSI，每芯片一百万晶体管。很多人以为五代机主张“非–冯”架构，其实一开始他们还真是把“冯”架构的硬件

改进也算到里面的。

1981 年在日本信息处理开发中心（JIPDEC）召开了第一届第五代计算机会议，对外公布元冈达委员会的报告。会议邀请了外籍嘉宾，其中有专家系统鼻祖、斯坦福大学的费根鲍姆和机器定理证明元老、德国人白贝尔（Wolfgang Bibel）。费根鲍姆的特约演讲强调了知识重于推理的理念，他还是念念不忘他的老师司马贺和定理证明中逻辑派结下的梁子。而白贝尔自然还是强调逻辑的重要性。这俩人基本是各说各话，但费根鲍姆的关键词“知识”和白贝尔的关键词“逻辑”都是日本人最想听到的，他们倒并不太关心欧美之间“两条路线”的斗争。关于“知识”和“逻辑”的更加公正的说法是语义和语法之争，知识是语义，而逻辑是语法。定理证明是纯粹依靠语法之力，但加入语义确实会提高效率。

相比于新的逻辑程序语言 Prolog，LISP 是更成熟的函数式编程语言，也是美国人工智能研究者的标准语言。日本电讯 NTT 的武藏野实验室（Musashino Lab，相当于美国的贝尔实验室）也更倾向于 LISP。最后的决定权落在野心勃勃的渕一博（Kazuhiro Fuchi）手中，他是当时第五代计算机研究所（ICOT，Institute of New Generation Computer Technology）的所长。他在会议上发表的论文明确强调了逻辑程序和 Prolog 的重要性。英国人沃伦（David Warren）1982 年在 *AI Magazine* 上发表的论文分析了渕一博为什么选择了 Prolog：在五代机的所有应用中，Prolog 能覆盖的面最广。五代机的终极目标是知识信息处理，在当时的语境下特指专家系统和自然语言理解。那时的自然语言理解还是规则为王的时代。Prolog 很明显比 LISP 更适合这些应用。而且在小规模数据的基准测试中，Prolog 作为数据库查询语言甚至不输关系代数。当然，Prolog 不是美国人发明的，这也是一个重要因素，这让日本人更有机会掌控并占据领先地位。日本人的民族自信

在这里起着隐形的作用。

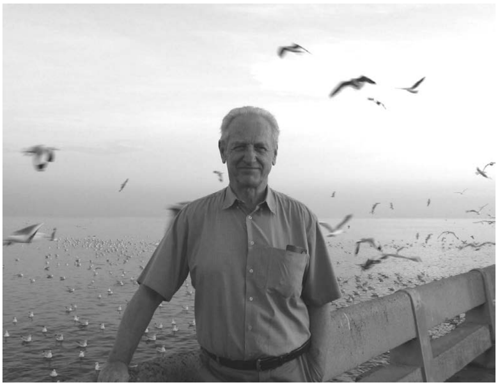

渕一博（1936—2006）

日本人是何时最早接触 Prolog 的也是很有意思的故事。古川康一（Koichi Furukara）被派到美国斯坦福研究所（SRI）做访问学者，他在那里的指导老师是巴罗（Harry Barrow）。巴罗正在自学 Prolog，他刚从沃伦手中买了 Prolog 解释器，但又没时间玩，于是把这个任务交给了新来的古川康一。古川很快就把 Prolog 在 SRI 的一台 DEC-10 上跑通了。他回国后就成了日本最早有 Prolog 实战经验的专家，马上进入刚成立的五代机研究所，并很快升为副所长，曾一度代表五代机研究所与欧洲的大学和研究机构周旋，企图把更多的欧洲人拉到日本的阵营里。

2. 理论基础：逻辑程序和 Prolog

在某种意义上，逻辑程序（Logic Programming）是自动定理证明“歪打没正着”的副产品。定理证明在归结原理之后经历了各种改进，仍然无法避免中间子句指数爆炸式的增长。进入 20 世纪 70 年代，人们开始研究证明模式，首先想到了线性归结，即整个定理证明过程沿着一条主线（中央子句），这样的优点是证明过程是目标制导的，有点像人的证明过程。

在英国爱丁堡工作的美国人科瓦尔斯基（Robert Kowalski）1971 年发明了 SL 归结并证明了 SL 归结的完备性，SL 归结就是线性归结的一种。这一方面奠定了科瓦尔斯基在定理证明界的地位，同时又让他有机会开创了逻辑程序。



科瓦尔斯基（1941—　）

科瓦尔斯基的早期学术生涯就一个字：乱。他本科先是在名校芝加哥大学，但两年后不适应那里的环境，回到老家，在三流学校桥堡大学（Bridgeport）拿了学位。后来再到名校斯坦福大学读数学博士，但两年后他再次离开。有两方面原因导致他出走，一是政治，二是学术。他曾经驾机企图到玫瑰碗（Rose Bowl，美国大学橄榄球决赛）的上空撒反战传单，只因飞机故障，飞到玫瑰碗上空时，球赛已结束，最后把传单都撒到了附近的迪斯尼乐园。尽管定理证明的关键工作都是在美国做的，但貌似欧洲对定理证明的兴趣却远高于美国。

科瓦尔斯基曾说："作为学生，我爱逻辑，恨递归论。"（As a student, I loved logic and hated recursion theory.）这也算是欧美之分吧。数理逻辑可分为一阶逻辑以及四大论——模型论、集合论（包括高阶逻辑）、递归论（也就是可计算性理论）和证明论。递归论当然也属于逻辑，只不过科瓦尔斯基更喜欢一阶逻辑。他 1967 年来到英国 AI 重镇爱丁堡大学，那里有数学

家梅尔泽（Meltzer）研究数学机械化，还有战时跟随图灵在布莱彻利庄园破译德军密码的米奇（Donald Michie）。米奇战前本科读的是经典学，战后读了个哺乳动物遗传学的博士，早期干 AI 的有生物背景的真不少。当时在爱丁堡的学生还有邦迪（Alan Bundy）、波尔[①]（Boyer）和摩尔（Moore）、海思（Patrick Hayes）等。科瓦尔斯基跟随梅尔泽学习定理证明，刚一到就碰见了正在那访问的罗宾逊。随后他的学术生涯一帆风顺。

科莫饶尔（Colmerauer）是马赛大学人工智能小组的头儿，当时正在研究自然语言问题回答系统，他邀请科瓦尔斯基访问马赛，讨论如何用 SL 归结做问题回答。科莫饶尔和他的学生卢梭（Roussel）开始考虑设计一个全新的程序设计语言，卢梭的老婆想到了 Prolog 这个名字，意为“逻辑编程”（Programming in Logic）。卢梭建议用 Horn 子句。Horn 子句就是 A1 and A2 and ... and A*n* → B 的格式。这里所有的 A 都不是负文字，转换成只有“或”连接词的子句就是 ~ A1 or ~ A2 or ... or ~ A*n* or B, 也就是说，一个子句里最多只能有一个正文字。科瓦尔斯基一开始并不满意 Horn 子句，因为这明显降低了表达能力。最后还是勉强接受了，一方面是为了效率，另一方面是大弟子沃伦的抗议，他本来派沃伦去研究 Horn 子句的短版，却不想沃伦反水，变成 Horn 子句的支持者。卢梭用 Algol-W 实现了第一个 Prolog。沃伦后来发明了沃伦抽象机（WAM），为后来的 Prolog 实现提供了基础。科瓦尔斯基此时开始写一本小册子《问题求解的逻辑》，这本册子越写越厚，1979 年成书出版后成为逻辑程序设计的“圣经”。后来他进一步提出了“算法=逻辑+控制”的口号，这明显是在学 Pascal 发明人沃斯（Niklaus Wirth）提出的“程序=算法+数据结构”的说法。一门学科没

① 波尔最后回到得克萨斯大学奥斯汀分校得了博士，他的第一个学生就是后来把吴文俊方法发扬光大的周咸青。

个能叫得响的口号恐怕火不了。这样的口号还很多，例如逻辑学家比森（Michael Beeson）在评论机器定理证明时就有“数学=逻辑+计算”之说，逻辑是表达，计算是过程；逻辑是几何，计算是分析。更广义地说，定理证明是逻辑，计算机代数（包括几何定理证明）是计算。

至于 Horn 子句的表达能力问题，早在 1961 年就有几位逻辑学家证明过 Horn 子句可以计算所有递归函数，即使像 Prolog 这样不完全的 Horn 实现仍可计算所有递归函数。其实还有比 Horn 子句更约束的形式，例如 Datalog。在 Datalog 中，每个项（term）可以是常量或变量，但不能是函数。有一个很简单的证明：Datalog 等价于关系数据库+递归。如果 SQL 允许递归，那么就已经实现了 Datalog。其实 SQL 每张表都可表示为 Datalog 的一个谓词，表中的一个列对应于谓词中的一个变量。

1975 年，科瓦尔斯基带着几位弟子跳槽到了伦敦的帝国理工学院，把那儿搞成逻辑程序设计的大本营。日本五代机计划决定以 Prolog 为基础之后，科瓦尔斯基一下成为被关注的中心，各种时尚类杂志也请科瓦尔斯基做封面。他也顺水推舟地为后来的日本五代机添油加醋。

3. 五代机计划和五代机研究所

日本通产省（MITI）大概相当于中国原来的信息产业部，稍带科技部加中科院，2001 年重组为经济产业省（METI），大概相当于中国现在的工信部。就像所有其他东亚国家，日本政府对私营企业有重大影响力。通产省决定了日本的科技和工业政策，同时，因为日本和欧美有巨大文化差异，通产省还起着翻译与中介的作用。所有这些权力加在一起，自然说一不二。另外，日本政府的战略考虑相对长期，一个项目至少要考虑十年计

划。这和美国政府受选举影响的短视行为形成鲜明对比。越战后，美国政府的资助机构已鲜有长期计划。

日本 MITI 对五代机的自信来自 DRAM 存储芯片的成功，20 世纪 70 年代日本半导体工业在 MITI 的协同下，组织了业界协会，在很短的时间内，DRAM 研发全面赶超美国，日本在计算机硬件制造方面由此对美国构成威胁，MITI 不满足于跟随美国，产生更大的野心，要在整个 IT 领域设立自己的标准。ICOT 选定 Prolog 而不是 LISP，一个主要原因就是 LISP 是美国制造，而 Prolog 是相对全新的。

通产省为五代机制定的十年计划是四亿五千万美元，第一期头三年通产省独资四千五百万到五千万美元，后两期参与的公司会有一比一匹配资金。所以总投入预期会到八亿五千万美元。而日本历来工业界对政府项目的匹配都会超过政府出资，所以通产省的小算盘是整个项目有可能达到十亿美元投入。虽然照美国标准，这不是巨资，IBM 1982 年一年的研发经费就是十五亿美元，但在当时的日本，这已经是史无前例了。

通产省之前也不是没有资助过这个规模的项目，超级计算机和新材料等项目都花钱不少。但那些项目属赶超而不是创新型项目，赶超是“有人地”，结果相对可预测，通产省从没失过手。但创新型项目是“无人地”，创新的本质就是谁都不能预知结果。

通产省内的电子技术综合研究所（ETL，Eletrotechnical Laboratory）是其最大的电子和计算机研究机构，大概类似中科院电子所加计算所加自动化所。现在 ETL 已和通产省其他研究所合并变成了“产业技术综合研究所”，从东京城里迁往筑波科学城。ETL 的渕一博年轻有为，是 ICOT 所长的不二人选。他的任务就是从 ETL 和参与的所有公司中挑选出最优秀的 40 名年轻研究者加入 ICOT。渕一博 1961 年 26 岁时就访问过伊利诺伊大

学，参与过超级计算机 Illiac II 的研发，熟悉美国研发文化。此时年仅 46 岁的渕一博要求他的所有部下不能超过 35 岁，他这是在刻意纠正日本文化尊老的习俗。在日本幕府时代，离开主家而流浪的武士被称为“浪人”。渕一博把他的 40 位部下戏称为“四十浪人”。这 40 个人加入 ICOT 后，虽然不会担心终身雇用，但他们在原单位的升迁机会没有了，从这个意义上说他们是浪人一点也不为过。

费根鲍姆受邀参加第一次五代机会议。他到达东京的当天，恰逢日立和三菱的几位高管因为窃取 IBM 的商业机密被判刑。美国那时开始注意到来自日本的潜在威胁，并采取行动。费根鲍姆观察到日本人的复杂心理：一方面他们认为这是美国对日本的贸易战；另一方面，他们也为此事羞耻——自己做不出来也不该偷。五代机给日本人提供了一个弯道超车的机会，日本同行认为有了五代机，就可以在信息产业里甩掉山寨的帽子。日本在软件方面大约落后十年，但如果五代机能够成功，那将是日本后发制人的又一案例。比较当时的日本和当下的中国，倒是有各种相似：被美国羡慕嫉妒恨、在制造业有长足进步、钢产量第一、能造高铁、那时的 Walkman 大概相当于现在的手机。就像当下各种民族主义的“说不”或“不高兴”，那时恰是原创《日本可以说不》盛行的时代。

渕一博的新办公室在东京的山寨性地标建筑东京塔（Tokyo Tower）的边上，面对东京湾，窗户下面就是 1853 年敲开日本国门的美国人佩里（Matthew Perry）停泊那几艘著名的黑船的地点。渕一博在初次见到费根鲍姆时透着民族骄傲，但费根鲍姆察觉到渕一博张扬外表掩盖的自卑。1854 年，佩里准将带来了更多的军舰。此时的日本已经知道他们的中国邻居鸦片战争大败，反抗不平等条约是不可能的，但幕府的保守派为了炫耀武力，安排佩里观看了相扑。作为答谢，佩里送给日本人按四比一比例缩小建造

的蒸汽机车和 100 米轨道，供官员和百姓乘坐。正是这个蒸汽机车彻底摧毁了日本武士们残存的义和团式自尊，打开了日本国门。一个有意思的细节是佩里 1853 年旗舰上悬挂的只有 31 颗星（那时美国只有 31 个州）的国旗 92 年后被海军上将哈尔西（William Halsey）借用，挂在密苏里号巡洋舰上接受日本投降。费根鲍姆对日本和五代机的洞察部分得自于他的日本裔太太 Penny Nii，她从费根鲍姆的知识系统实验室得了博士，一直都没离开专业。

4. 并发 Prolog

计算机科学的进步主要是由两件事驱动的：机器太慢；写程序太慢。在任何时间点，在半导体技术的约束下，提高机器速度的唯一办法是并行处理。Prolog 是高级语言，表达效率肯定要比过程语言更高。于是，如何并发地执行 Prolog 在五代机项目一开始就是重要的研究课题。科瓦尔斯基说逻辑程序设计是知识处理和高度并行体系结构之间的可能联系（missing link）。

串行 Prolog 的语义是由科瓦尔斯基定义的，例如，如果要证明 A，在规则库中有

$$A \leftarrow B_1 \ \& \ B_2 \ \& \ \ldots \ \& \ B_n.$$

那么就一个个地去证明 B_i。如果所有的 B_i 可以并行地归约，那么就是 AND 并行性。当有多个子句可以和 A 合一，那么又可以让并行性在多个子句中发生，这就是 OR 并行性。

帝国理工学院的 Clark 和 Gregory 提出的关系语言（Relational Language）是最早的逻辑编程语言的并行方案。1982 年刚刚从耶鲁大学博士毕业的夏

皮若（Ehud Shapiro）到以色列魏茨曼理学院做博士后，他参加了 1981 年的第一次五代机研讨会，在到五代机研究所访问时，受关系语言的启发，提出了 Concurrent Prolog 语言。在 NEC 工作的上田和纪（Kazunori Ueda）在 Concurrent Prolog 的基础上提出了 Guarded Horn 子句 GHC。这个被夏皮若戏称为“日本制造”的发明迅速成为五代机的核心语言 KL1。而夏皮若自己后来则走了另一条路：Concurrent Prolog 的 AND 并行性的子集 Flat Concurrent Prolog。夏皮若后来用 Concurrent Prolog 写了个以色列的“微信” Ubique，后来卖给了 AOL。五代机失败后，夏皮若头脑灵活，改行做生物信息学和分子计算机，现在是以色列计算机科学的大佬。

并行 Prolog 也带动了 LISP 并行机的研发。1983 年从麻省理工学院分出来的创业公司 Thinking Machines 成立，它的域名 think.com 是第三个.com 域名，创始人希利斯（Danny Hillis）的三个导师都大名鼎鼎：香农、明斯基和萨斯曼（Sussman）。Thinking Machines 当时是美国最火的创业公司之一，经费来源主要是美国政府——并非投资而是客户。随着日本五代机的失败，Thinking Machines 也失去了战略作用，遂在 1994 年关门。希里斯随后去迪斯尼研究哲学，顺便开了家公司 Metaweb，不多久就被谷歌收购变成谷歌的开源知识图谱 Freebase，而 Metaweb 的 CTO John Giannandrea 则成了谷歌 Xlab 的头，现在升为谷歌搜索的主管。无独有偶，MCC 的最后有价值的资产 Cyc 恰是最早的知识图谱。但这些只能算是非物质文化遗产，和五代机的初衷关系已经不大。

当时在美国，在硬件方面最接近日本五代机的研究工作是大卫・肖（D. E. Shaw）在哥伦比亚大学做的 Non-Von（非–冯）并行机，Non-Von 旨在用硬件实现知识表示语言 KRL。但在 1986 年五代机如日中天的时候，大卫・肖离开学界加入摩根斯坦利，1988 年创办了投资公司 DE Shaw，成

为华尔街大鳄。功成名就后，他从 DE Shaw 半退休，又办了 DE Shaw 研究所，从事生物化学和制药的研究，核心工具仍然是并行处理。

5. 美国和欧洲对日本五代机计划的反应

美国的回应

20 世纪 50 年代，美国在苏联卫星上天的压力下，大力投资科技和教育。但当冷战进入缓和期后，美国人失去了以举国之力打造超级项目的动力。美国众议院贸易委员会在分析了美日贸易数字之后说美国对日贸易，除了飞机外，几乎就像一个发展中国家对发达国家的贸易，美国向日本提供农产品和原材料，日本向美国提供工业品、家用电器和高科技产品。其实当下的中美贸易也类似。

贝尔纳在《科学的社会功能》中指出，英国把他们最好的人才浪费在古典学（希腊语、拉丁语）上了，而现代社会需要大科学，需要最好的人才。在贝尔纳时代，科技反超的代表是苏联。20 世纪 80 年代，日本在制造业和集成电路大举超越美国时，大家再次反思美国的教育，认为应该向日本学习。

太多的律师和会计师是对整个社会资源的浪费。美国的律师协会曾经抱怨说日本对外籍律师的签证额度限制导致美国律师所没法在日本干活。但日本人不鼓励美国律师带来的诉讼习气，认为这会拖累整个日本对技术创新的追求。按人口平均算，日本的律师数量是美国的二十分之一，会计师是美国的七分之一，但工程师却是美国的五倍。费根鲍姆颇为认真地说应该简化美国的法律流程，这样就会有多余的律师和会计师，而美国应该设立成人教育计划，把这些多余的律师和会计师，连带反正也找不到

工作的英语博士训练成工程师。

但 20 世纪 90 年代五代机失败后，大家又各说各话。再后来，在政治正确性的压力下，美国和欧洲已经丧失了斗志。美国越战后愈演愈烈的反智主义阻碍了科技教育。全才作家布莱森（Bill Bryson）在其英国游记里观察到，当牛津附近的居民在为三星刚开的工厂能提供 800 个蓝领就业机会而欢欣时，牛津的孩子们还是在研读古典学，而不是数理化。近来，伴随着中国崛起，中国式“虎妈”教育又开始风行美国。

当时，所有的美国商学院都用日本索尼 Walkman 的成功作为案例，想说明日本会占据技术市场。把美国过去几十年的商学院案列教材拿出来翻翻倒是蛮有意思的，那些当时貌似惊人的案例，没过多久就都完蛋了。商学院对日本的研究的另一个流行观点是，日本的终身雇用制让公司和雇员都对公司的发展采取长期的视角，但现在日本已开始改变终身雇用制。商学院教学的信用完全是建立在人们的健忘上。

费根鲍姆则把五代机当作日本人的“新政”。我觉得费根鲍姆是“勾结”境外“反动”势力（日本人），给本国政府施加压力，要求增加在科技领域，尤其是在自己领域的投资。他在日本的每一次出镜都为他在本国捞足了资本。最后无论是科研经费还是开公司，他都挣了不少钱，还落下一图灵奖。公司最终其实都被他搞黄了。在风险投资大行其道之前，美国创业者从政府拿钱是王道：既不稀释股份，又没有业绩压力。

日本的五代机项目为美国敲响了警钟。1982 年美国政府决定成立 MCC（Microelectronics and Computer Consortium），作为对日本五代机项目的回应，每年投资七千五百万美元，共 600 个职位编制。美国海军上将英曼被任命为 MCC 的董事长兼 CEO，英曼曾任美国国家安全局局长和美国中央情报局副局长。MCC 是一个工业界的松散耦合联盟（Consortium），

除了 IBM 和 AT&T 之外的美国所有重要高科技公司都参与。这么多公司联合办公，在美国历史上还是头一次，国会特批免除“反垄断法”的限制。很多批评者认为免除“反垄断法”涉及原则问题。但 20 世纪 50 年代末美国在苏联压迫下开始阿波罗计划的时候，也是搞政府协调，那时似乎并没有人用“反垄断法”说事。这真是一个反讽：原本为了提高工业界竞争力的“反垄断法”竟然限制了美国作为国家的竞争力。

27 个州的 57 个城市参与竞标 MCC 的选址，其中有加州的圣地亚哥、乔治亚州的亚特兰大、北卡的研究三角地，最后决定花落得克萨斯州的奥斯汀。硅谷的高管对此表示不满说：奥斯汀算哪根葱？（Where the hell is Austin？）但是英曼是得克萨斯人，并且毕业于得克萨斯大学奥斯汀分校。尽管奥斯汀并不以高科技闻名，得克萨斯大学奥斯汀分校的机器定理证明研究却是美国的重要基地之一，带头人是布莱索，他是计算机系的建系系主任。正是布莱索招募了波尔–摩尔证明器的开发者波尔和摩尔加盟。他是英曼的智囊之一，英曼一到任，就提名他为负责人工智能的副总裁。也恰是布莱索把雷纳特（Douglas Lenat）网罗到 MCC，并大力支持他的 Cyc 项目。

除了 MCC，DARPA 还建立了另外三个国防项目：无人驾驶车、飞行员辅助系统和战场管理系统（Battle Management System），里面有和机器–指挥官的自然语言理解界面。DARPA 无人驾驶车项目是现在各种类似项目的源头。

英国的阿尔维计划

沃伦在 1982 年评论刚出锅的日本五代机计划时说，日本和美国的类似计划的源头都是欧洲，更确切地说是英国。他大概忘了他的导师科瓦尔斯基实际是从美国逃到英国去的。英国政府于 1982 年夏婉拒了日本邀请联

合开发五代机的倡议，宣布将在未来五年内投入两亿五千万英镑（等值三亿六千万美元）开发英国人自己的阿尔维计划（Alvey Program）。作为对比，同年的马尔维纳斯群岛之战英国花了七亿英镑。英国撒切尔政府面临压力要求阿尔维计划必须产生可市场化的产品。1987 年在日本五代机计划进入胶着状态时，英国宣布放弃阿尔维计划。英国人似乎对失败超级敏感，退出总是很快。事后，阿尔维计划的头儿奥克利（Brian Oakley）1991 年在《科学》杂志上撰文马后炮自抽脸："把繁荣寄望于研究，英国也真是够蠢的。"

战时跟随图灵在布莱彻利庄园破译德军密码的米奇当时是爱丁堡大学人工智能单位（Artificial Intelligence Unit）的头儿，这个人工智能单位当时是一个学院的编制。米奇 1983 年在人工智能鼎盛时期离开爱丁堡，去不远处的格拉斯哥建立了图灵研究所（Turing Institute），1994 年没钱关门，整个历程和日本的五代机的时间点重合。2015 年英国政府联合了五所大学（剑桥、牛津、爱丁堡、UCL 和华威大学）再次成立阿兰·图灵研究所，新的研究所坐落于大英图书馆内，定位大数据。看起来每次有新的话题时，图灵都会"复活"一次。

欧洲的 ESPRIT

1978 年，整个欧洲经济共同体在信息技术有 50 亿美元贸易顺差，但 4 年后的 1982 年却有 120 亿的逆差。日本五代机项目更加重了紧迫感。1983 年欧洲启动了"欧洲信息技术战略计划"（ESPRIT），十年预算是十五亿欧洲货币单位 ECU（1999 年改为欧元）。劳工的短缺也是上马 ESPRIT 的原因。随着 ESPRIT 逐步退火，大家又说机器靠不住，还得靠人。不知人工智能技术会不会改变欧洲的人口结构。

法国的总统们都特有文艺范儿，前有德斯坦生活浪漫，后有密特朗

请贝聿铭在卢浮宫门口搭建玻璃三角。密特朗想当信息时代的狄德罗——做电子百科全书，而这本来在日本的五代机项目中也有规划。于是法国建立了世界信息技术和人力资源中心（World Center for Information Technology and Human Resources），他们当时请的首席科学家是明斯基的战友、2016 年过世的佩珀特（Seymour Papert）。但说起电子百科全书，最终还是民主的维基百科成了气候。

德国更加务实些，1988 年在别国已经开始对五代机采取谨慎态度时，他们上马了德国人工智能研究中心（DFKI）。相对于其他浮云，DFKI 今天还在，在经历了人工智能的一个循环之后，仍然是欧洲 AI 的中心。

6. 结局和教训

五代机的衰落在 1988 年就已露出端倪。1981 年的第一次五代机会议的会议录只有 280 多页，其中还包括 89 页的元冈达报告。而 1988 年的五代机会议录有 1300 页三大卷。但从会议录中可以看出，五代机已经成了大杂烩，失去了聚焦点，给人感觉五代机不会在任何相关领域取得突破性进展，而同时八杆子打不着的领域也拼命向五代机靠拢。就像当下人工智能领域的创业，一些和人工智能毫无瓜葛的社会闲杂人员一夜间都成了 AI 专家，拿到 VC 的钱之后再想干啥。

五代机没有能证明它能干传统机不能干的活，在典型的应用中，五代机也没比传统机快多少。休伊特（Carl Hewitt）是麻省理工学院的教授，明斯基和佩珀特的学生，早年的工作 Planner 对自动定理证明有很大影响。他认为大部分五代机的工作都是试着用逻辑程序去解决其他手段早就解决的问题，而不是去解决其他手段不能解决或解决得不好的问题。他还认

为逻辑编程和并行性天生就是一对不可调和的矛盾。

20 世纪 80 年代，日本经济增长率在 4%，但到了 90 年代则降到 1%。在讨论五代机时，很少有人从日本国情和经济角度研究，但在分析美国的问题时，都会讲国情，出了美国却只能讲技术和行业。主要原因是大家对美国比较熟。

MCC 国际合作总监伊顿（Eaton）并不认为五代机失败了：MITI 对五代机的投入并不大，五代机促成了 20 世纪 80 年代中后期 AI 的繁荣，也确实提升了日本在全世界的形象。日本人工智能学者甚至说如果有现在的语义网和知识图谱的大数据，五代机结局会很不同。但历史没法马后炮。

五代机的后期赶上互联网的突起，相比互联网，五代机自然光辉不再。五代机的影响局限在技术圈内，而信息高速公路的影响却是全社会的。也许五代机并非是失败，只是各种技术此起彼伏的一个阶段。1994 年五代机项目结束后，渕一博转往东京大学任教，两年后退休。元冈达（1929—1985）和渕一博（1936—2006）都不算长寿。

7. 日本还有机会吗：日本下一代人工智能促进战略

日本总务省在 2016 年 7 月发布了“下一代人工智能促进战略”。这个新计划的目标是帮助日本克服日益严重的社会问题：人口老龄化和劳动力不足。20 世纪 80 年代，日本五代机的发起单位是通产省内的电子技术综合研究所（ETL），通产省现在已经变成了经产省，而 ETL 也几经重组变成了产业技术综合研究所，仍是日本“下一代人工智能促进战略”的核心单位。此时的日本已经没有了 30 年前力图超越美国的豪情壮志，总务省的报告承认日本在大数据和机器学习等新前沿，无论人才还是技术，已经

全面落后欧美。谷歌收购 DeepMind 对日本刺激很大，日本的大企业是不会收购没有什么营收的公司的。总务省认为日本应该以国家为中心，为从事人工智能研发的合资企业和大企业搭建平台。如果日本在人工智能领域的国际竞争力被进一步削弱，日本产业在未来将面临衰败。报告骄傲地提到了 NEC 的普林斯顿研究所为人工智能产业培养了一批人才，还提到日本早年从事神经网络研究的先驱福岛邦彦（Fukushima）①和甘利俊一（Amari），旨在鼓励日本从事人工智能的研究者。但事实上，在福岛邦彦和甘利俊一的壮年，日本把资金都投入了五代机，他们没赶上好时候。

参考文献指南

Moto-Oka (1982)是日本第五代计算机项目的宣言，标志着日本这一雄心勃勃项目的开始。费根鲍姆在项目的早期被以“外来的和尚”的形式请到日本，他和他的御用作者麦克达克写了本描述五代机项目早期内幕的书，见 McCorduct and Fengenbaum (1984)，但这本书更多地旨在唤醒美国的科技政策决策者，当然也是为自己能拿到更多政府资助奠定舆论基础。

第五代计算机的历届学术会议的论文集是很好的参考读物，例如，Furukawa (1987)、ICOT (1988)、Kruozumi (1992)，等等。

相关的逻辑程序设计的学术研究尽管还在继续，但早已失去当年的火爆劲头。Kowalski (1986)和 Kowalski (1988)是早期历史的第一手资料，科瓦尔斯基退休后对逻辑程序设计多有反思，Sergot (2002)是对科瓦尔斯基生平和学术很好的回顾。逻辑程序设计的一些思想在后来的知识图谱研究中也有体现。

---

① 当下流行的卷积神经网络 CNN 的源头就是福岛邦彦的工作。

第 5 章
神经网络简史

I bet the human brain is a kludge.

人脑就是一台计算机。

——Marvin Minsky（明斯基）

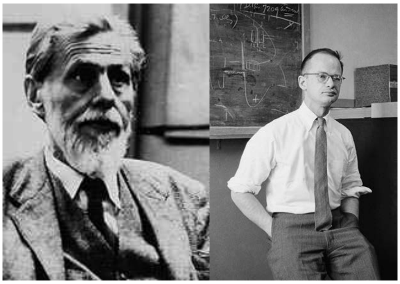
自图灵提出“机器与智能”起，就一直有两派观点：一派认为实现人工智能必须用逻辑和符号系统，这一派看问题是自顶向下的；还有一派认为通过仿造大脑可以达到人工智能，这一派是自底向上的，他们认为如果能造一台机器，模拟大脑中的神经网络，这台机器就有智能了。前一派，我想用“想啥来啥”来形容；后一派就称之为“吃啥补啥”，估计他们的思想来源于中国古代的原始思维，套一句庸俗的哲学词，前者偏唯心，后者偏唯物。这两派一直是人工智能领域里“两个阶级、两条路线”的斗争，这斗争有时还你死我活。

1. 神经网络的初创文章

模拟神经网络的原创文章发表于 1943 年，两位作者都是传奇人物：麦卡洛克（Warren McCulloch）和皮茨（Walter Pitts）。皮茨打小就喜欢数学和哲学，初中时就读过罗素的《数学原理》，还和罗素通过信。罗素爱才，邀请他到英国跟随自己学习逻辑。但皮茨是苦出身，连高中都读不起，

英国留学自然未果。他 15 岁时，他爸强行要他退学上班养家。就像所有爱读书的穷孩子，皮茨一怒之下就离家出走了。他打听到偶像罗素那时要到芝加哥大学任教，就只身来到芝加哥，还真见到了罗素。老罗遂把他推荐给那时也在芝加哥任教的卡尔纳普。卡尔纳普想看看这孩子到底有多聪明，就把自己的《语言的逻辑句法》一书送了一本给皮茨。不到一个月，皮茨就看完了，把写满笔记的原书还给卡尔纳普。老卡惊为天人，于是给他在芝加哥大学安排了一份打扫卫生的工作。别看不起打扫卫生，在电影《心灵捕手》（*Good Will Hunting*）里，马特·达蒙饰演的角色就是在知名大学打扫卫生时，不小心解了一道数学难题，引起了老师的注意。扫马路至少可以避免流浪街头。皮茨后来结识了也在芝加哥的麦卡洛克。麦卡洛克比皮茨大一辈，有人称他是皮茨的养父。麦卡洛克本科在耶鲁大学学习哲学和心理学，后在哥伦比亚大学获得了心理学硕士和医学博士（MD）学位。其实医学博士和哲学博士不是一回事，MD 不是学术学位，而是终极职业学位，和 MBA、MFA 差不多。MD 的那个“D”是指“医生”，PhD 的“D”才是博士。

麦卡洛克毕业后做了几年实习医生，先去了耶鲁大学研究神经生理学，后来又去了伊利诺伊大学芝加哥分校，做精神病学系的教授。麦卡洛克的强项是神经科学，但不懂数学，他和 17 岁的流浪汉数学票友皮茨是绝配。他们合作的成果就是神经网络的开山之作：A Logical Calculus of Ideas Immanent in Nervous Activity，发表在《数学生物物理期刊》上。这篇文章成了控制论的思想源泉之一。有意思的是，这篇文章只列了三篇貌似不相关的参考文献，第一是卡尔纳普的《语言的逻辑句法》，第二是希尔伯特和他学生阿克曼合著的《数理逻辑基础》，第三是怀特海和罗素的《数学原理》。



麦卡洛克（1898—1969）与
皮茨（1923—1969）

控制论的创始人维纳（Norbert Wiener）早年自称神童，他爸是哈佛大学教授，曾经带着他到英国见过罗素，但罗素特不喜欢这孩子和他爹。自打进入 20 世纪后，甭管哪门哪派的学问，最后都能扯到罗素那儿，不想得诺贝尔文学奖的科学家不是好情人。维纳后来也在哈佛大学任教，但不被主流数学家喜欢，没拿到终身教职。最后到了隔壁的麻省理工学院落脚，在二战时搞了点武器研究。那时最好的数学家和物理学家都参与了造原子弹的“曼哈顿”计划，维纳却没沾边。这也许同他的个性有关系，他的同事和家人都觉得他对数学之外的事情反应迟钝。维纳提出“控制论”后出了大名，在麻省理工学院搞了一大笔钱，麦卡洛克就带着皮茨等一票人马投奔维纳。有钱才能当老大，哪都一样。



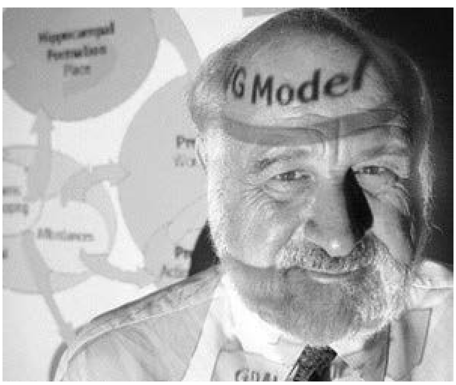
维纳（1894—1964）

维纳的老婆玛格丽特是纳粹，在二战时，家里还偷藏了本英文版的希特勒的《我的奋斗》。那时他们的女儿芭芭拉正在读小学，有意无意地也看过那书，写作文时居然引用书里的“警句”，差点被学校开除。麦卡洛克的老婆是犹太人，与玛格丽特形同水火。其实维纳祖上是波兰犹太人，玛格丽特早干啥去了？维纳娶玛格丽特是为了自嘲吗？就像很多中国男人讨洋老婆或老外娶中国剩女，或许图的不是相貌，是稀罕。反正最后维纳被中和为“不可知论者”（agnostic）。玛格丽特有次对维纳说麦卡洛克小组有人（可能暗指皮茨）勾引宝贝女儿芭芭拉，维纳大怒，随即断绝和麦卡洛克及其学生的所有往来。现在看玛格丽特是有意造谣。但维纳的举动对皮茨造成了巨大的创伤，皮茨本来是维纳的特招学生（special student），但估计他年幼时受过挫折，秉性怪异。和维纳闹翻后，他拒绝麻省理工学院给他的研究生学位，对学问也心灰意冷。1969 年，皮茨比他的长辈麦卡洛克早几个月离世，年仅 46 岁。

维纳曾写过两卷本的自传：《昔日神童》（*Ex-prodigy*）和《我是数学家》。不喜欢维纳的人开玩笑说，应该是《昔日数学家》和《我是神童》，嘲讽维纳的数学不入主流，同时暗示维纳对自己神童身份的过高自视。和维纳相熟的人，无论朋友还是敌人，都认为维纳的神童光环害了他，使他终身没有自信。维纳和麦卡洛克失和的另一个原因是他们迥然不同的学术风格。维纳无论如何首先是一位严谨的数学家，而麦卡洛克则被人称为是浪漫的科学家。所谓“浪漫”不是指生活，而是说他对科学思想的表述方式。维纳曾经把为大脑建模作为他学术生涯的最后野心，他曾经把麦卡洛克找来，要他指出脑科学最关心的几个问题，然后维纳自己花了两年时间把这些问题数学化，并试图给出解决的思路，但当维纳在一次生物学的会议上宣布自己的成果时，生物学家觉得维纳是胡闹，于是维纳觉得是麦卡

洛克给他下了套。

得维纳真传的人不多，不能不说一下阿比卜（Michael Arbib）。他 23 岁就在维纳手下得了博士，算是维纳最后一个学生。阿比卜本是英国犹太人，他爸二战时当兵被俘，战后举家迁到澳大利亚。他在悉尼大学读数学。他回忆大学时博览群书，而对他影响最大的是维纳的《控制论》、麦卡洛克和皮茨的神经网络、拉宾和斯考特的有限自动机，以及麦卡锡和香农编辑的文集《自动机研究》（*Automata Studies*）。他认真读过图灵的经典论文"可计算的数"，自称曾挑出过 31 个错，还翻译了哥德尔 1931 年那篇改天换地的文章[①]。他选择到麻省理工学院读博士，因为那里有维纳、麦卡洛克、皮茨，还有麦卡锡和明斯基。阿比卜到麻省理工学院时，维纳和麦卡洛克已经失和。尽管维纳是他名义上的导师，却很少提供指导，而他实际上花了更多时间和维纳的对头麦卡洛克在一起。他在拿到博士后才告诉维纳，维纳大怒。阿比卜曾如此评论维纳："伟人，但有人格缺陷。"[②]



阿比卜（1940—　）

① 哥德尔那篇文章直到 20 世纪 60 年代才被翻译为英文，被收录到逻辑学家海因诺特编辑的文集《从弗里格到哥德尔》。见尼克《哲学评书》。

② "great man but perhaps a defective human being."。

阿比卜虽是维纳的学生，但他并没有把自己局限于控制论的狭隘圈子里。他是全才，出版过计算理论、人工智能等多种专著及科普读物，甚至还一度玩过高深莫测的范畴论。和计算理论相比，控制论更不纯粹。阿比卜的杂学体现在他那本科普书《大脑、机器和数学》里，其实他本科毕业论文已露端倪，那篇题为 Finite Automata, Turing Machine, and Neural Networks 的文章发表在美国计算机学会会刊 *JACM* 上。阿比卜后来创办了麻省大学的计算机系，并延揽一帮人工智能人马，其中有后来以“强化学习”出名的巴托（Andy Barto），使麻省大学的人工智能曾在很长一段时间都处于领先地位。20 世纪 80 年代末期，阿比卜离开麻省大学，转往南加州大学，曾一度风光，担任一堆系（包括计算机、生物、生物医学工程、电气工程、神经科学，还有心理）的教授。他那名片要是印出来，估计会像一些农民企业家的那样长吧。但阿比卜最终并未成为开天辟地的宗师，有愧于他的天分和才华。南加州大学并没有因为他的到来而添彩，但麻省大学却因为他的出走而失去了自己的特色和主心骨。就像遗传算法的祖师爷霍兰德所说：自己的影响力很大程度上要看有没有出名的学生，学生是学术圈生态环境的一个环节。麻省大学有阿比卜需要的生态环境，南加州大学则有名无实。阿比卜晚年为自己日渐衰落的学术影响力找过借口，他认为原因是马尔（David Marr）学派的当道。马尔和他那一小撮把他当神一样崇拜的学生曾经一度统治了视觉研究领域，马尔的早逝加剧了马尔的神话，马尔的书《视觉》也成为学生的“圣经”。阿比卜认为马尔的工作是建立在自己的工作之上的，但“圣经”里并没提及，仿佛一切都是马尔自己一夜之间发明的。

1949 年，神经心理学家赫布（Donald Hebb）出版了《行为组织学》（*Organization of Behavior*）。在该书中，赫布提出了被后人称为“Hebb 规则”

的学习机制。该规则认为，如果两个细胞总是同时激活的话，它们之间就有某种关联，同时激活的概率越高，关联度也越高。换句话说，就是“吃啥补啥”。2000 年诺贝尔医学奖得主肯德尔（Eric Kandel）的动物实验也证实了 Hebb 规则。后来的各种无监督机器学习算法或多或少都是 Hebb 规则的变种。

2. 罗森布拉特和感知机

神经网络研究的后一个大突破是在 1957 年。康奈尔大学的实验心理学家罗森布拉特（Frank Rosenblatt）在一台 IBM-704 计算机上模拟实现了一种他发明的叫作“感知机”（Perceptron）的神经网络模型。这个模型可以完成一些简单的视觉处理任务。这在当时引起了轰动。



罗森布拉特（1928—1971）

罗森布拉特在理论上证明了单层神经网络在处理线性可分的模式识别问题时，可以收敛，并以此为基础做了若干“感知机”有学习能力的实验。罗森布拉特 1962 年出了本书《神经动力学原理：感知机和大脑机制的理论》（*Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*），

这本书总结了他的所有研究成果，一时成为“吃啥补啥”派的“圣经”。罗森布拉特的名声越来越大，得到的研究经费也越来越多。国防部和海军都资助了他的研究工作。媒体对罗森布拉特也表现出了过度的关注。毕竟，能够构建一台可以模拟大脑的机器，当然是一个头版头条的抢眼消息。此时的罗森布拉特也一改往日的害羞，经常在媒体出镜，他开跑车，弹钢琴，到处显摆。这使得另一派的人相当不爽。

明斯基是人工智能的奠基人之一，是达特茅斯会议的组织者。他在一次会议上和罗森布拉特大吵，认为神经网络不能解决人工智能的问题。随后，明斯基和麻省理工学院的另一位教授佩珀特（Seymour Papert）合作，企图从理论上证明他们的观点。他们合作的成果就是那本影响巨大、“是也非也”的书：《感知机：计算几何学》（*Perceptrons: An Introduction to Computational Geometry*）。在书中，明斯基和佩珀特证明单层神经网络不能解决 XOR（异或）问题。异或是一个基本逻辑问题，如果连这个问题都解决不了，那神经网络的计算能力实在有限。其实罗森布拉特也已猜到“感知机”可能存在限制，特别是在“符号处理”方面，并以他神经心理学家的经验指出，某些大脑受到伤害的人也不能处理符号。但“感知机”的缺陷被明斯基以一种敌意的方式呈现出来，当时对罗森布拉特是个致命打击。原来的政府资助机构也逐渐停止对神经网络研究的支持。1971 年，罗森布拉特在 43 岁生日那天划船时淹死。很多人认为他是自杀。王国维沉湖时遗言“经此世变，义无再辱”，而对于罗森布拉特，我猜“辱”是明斯基的书，“世变”是随后“神经网络”学科的消沉。不同的是，王静安谓之“世变”是历史潮流，但神经网络学科十年后会逆袭。

表面看是因为科学，但有证据表明明斯基和罗森布拉特以前就有瓜葛。他们是中学同学。布朗克斯（Bronx）科学高中大概是全世界最好的

高中，毕业生里得过 8 个诺贝尔奖、6 个普利策奖、1 个图灵奖。远的不说，明斯基是 1945 年的毕业生，而罗森布拉特是 1946 年的毕业生。美国高中学制四年，明斯基和罗森布拉特至少有两年重叠，而且彼此认识，互相嫉妒。1956 年的达特茅斯会议定义了“人工智能”这个词，会议的组织者包括明斯基、麦卡锡和香农等，参会者还有司马贺、纽厄尔等。这个会议在定义“人工智能”领域时只是提到了神经网络。那时明斯基是神经网络的支持者。他 1954 年在普林斯顿大学的博士论文题目是“神经–模拟强化系统的理论，及其在大脑模型问题上的应用”[①]，实际上就是一篇关于神经网络的论文。他晚年接受采访时开玩笑说，那篇 300 多页的博士论文从来没有正式发表过，大概只印了三本，他自己也记不清内容了。貌似他想极力开脱自己和神经网络学科千丝万缕的关系。达特茅斯会议的主题并不是神经网络，而是后来被纽厄尔和司马贺称为“物理符号系统”的东西，也就是说，“想啥来啥”派是主要基调。

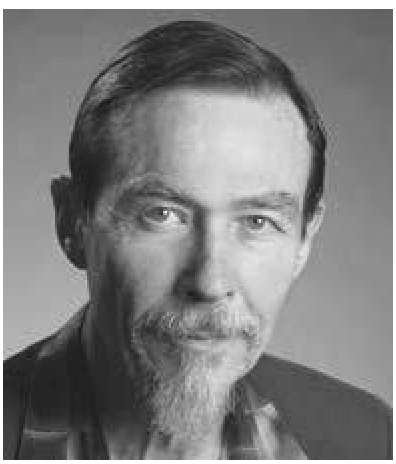
罗森布拉特被比他大一岁的明斯基妒忌是自然的。工作上，明斯基所负责的麻省理工学院的人工智能实验室也在向美国国防部和海军申请经费。大多数圈内的科学家对罗森布拉特突然被塑造起来的明星范儿很反感。明斯基早期也是“吃啥补啥”派出身，但此时已经改为“想啥来啥”派了。由于他和佩珀特对感知机的批判，俩人后来被“吃啥补啥”派称为“魔鬼搭档”。其实明斯基结识佩珀特还是通过麦卡洛克的介绍，历史真是纠结。被称为“魔鬼”是因为《感知机》第一版有言：“罗森布拉特的论文大多没有科学价值。”这话跳步确实有点大，但罗森布拉特人缘不好，没有得到同行的支持。

---

① Theory of Neural-Analog Reinforcement Systems and its Application to the Brain-Model Problem。

比罗森布拉特小一岁的维德罗（Widrow）是斯坦福大学教授，在罗森布拉特刚提出“感知机”时，他就提出了 Adaline 可适应性算法。Adaline 和感知机很相似，也是机器学习的鼻祖模型之一。罗森布拉特享受盛誉时，维德罗也沾了光，但在罗森布拉特死后，他却并没有被非难。维德罗在几十年后回忆说，那是因为他后来主要在电机系（EE）做集成电路的工作，而不是在计算机系里从事派系繁杂的人工智能研究，圈子不同，老死不相往来。

感知机的失败导致了神经网络研究的式微，用加州理工学院的集成电路大佬米德（Carver Mead）的话说是“二十年大饥荒”。明斯基在《感知机》一书再版时，删除了原版中对罗森布拉特个人攻击的句子，并手写了“纪念罗森布拉特”（In memory of Frank Rosenblatt）。但其他在“大饥荒”时期受到压迫的科学家认为明斯基不可原谅，后来神经网络得势后，这些人纷纷对明斯基口诛笔伐。美国电气电子工程师协会（IEEE）于 2004 年设立了罗森布拉特奖，以奖励在神经网络领域的杰出研究。



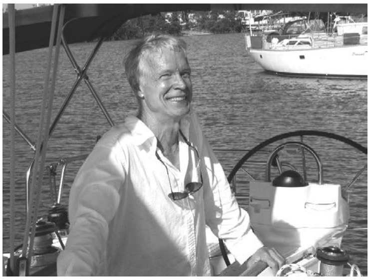
米德（1934— ）

3. 神经网络的复兴

在信息科学和神经科学的结合部的失败，并没有影响到神经生物学内部。哈佛神经生物学家胡贝尔（David Hubel）和威瑟尔（Torsten Wiesel）对视网膜和视觉皮层（visual cortex）中神经细胞的信息处理模式做了深入研究，他们为此获得了 1981 年的诺贝尔医学奖。随后，麻省理工学院的马尔为视觉信息处理建立数学模型，影响了后来连接主义的运动。威瑟尔后来离开哈佛大学去了洛克菲勒大学。1991 年，洛克菲勒大学时任校长的巴尔的摩出了学术丑闻被迫辞职后，威瑟尔出任洛克菲勒校长，为维持那所学校作为生物学重镇的地位做出了贡献。

1974 年，哈佛大学的一篇博士论文证明了在神经网络多加一层，并且利用“后向传播”（back-propagation）学习方法，可以解决 XOR 问题。这篇论文的作者是沃波斯（Paul Werbos），他后来得到了 IEEE 神经网络学会的先驱奖。沃波斯这篇文章刚发表时并没引起多少重视，那时正是神经网络研究的低谷，文章不合时宜。

神经网络在 20 世纪 80 年代的复兴归功于物理学家霍普菲尔德（John Hopfield）。1982 年，那时在加州理工学院担任生物物理教授的霍普菲尔德，提出了一种新的神经网络，可以解决一大类模式识别问题，还可以给出一类组合优化问题的近似解。这种神经网络模型后来被称为霍普菲尔德网络。1984 年，霍普菲尔德用模拟集成电路实现了自己提出的模型。霍老也培养了一批后起之秀，包括现在在生物学重镇 Salk 研究所担任计算神经生物学实验室主任的塞吉诺斯基（Terry Sejnowski）。霍老后转往普林斯顿大学担任分子生物学教授，现已退休。


霍普菲尔德（1933—　）

霍普菲尔德模型的提出振奋了神经网络领域。神经网络的这次复兴和生物学没啥关系，它既不是来自生物学的刺激，也没有给生物学送去任何慰藉。倒是它来源于物理学家，并引起了物理学家的关注，曾经一批对复杂系统感兴趣的物理学家在物理学家的交叉学科杂志 *Physica D* 上接二连三地发表文章，好不热闹。

一帮早期神经网络研究的幸存者，在生物学家克里克（Francis Crick[①]）和认知科学大佬诺曼（Don Norman）的鼓励下，以加州大学圣地亚哥分校为基地，开始了连接主义（Connectionism）运动，这个运动的领导者是两位心理学家鲁梅尔哈特（David Rumelhart）和麦克利兰德（James McLelland），外加一位计算机科学家辛顿（Geoffrey Hinton）。


辛顿（1947—　）

① 克里克就是发明 DNA 双螺旋的那位诺贝尔奖得主。

连接主义运动的成果之一就是那本被称为 PDP（*Parallel and Distributed Processing*）的著名文集（分两卷）。此书的出版给认知科学和计算机科学吹了股大风，被后起的神经网络新秀称为“圣经”。此书第一次印刷就印了 6000 本，这在科技文集类书里实属少见。20 世纪 80 年代的“神经网络”就像 20 世纪 90 年代的互联网、后来的 Web 2.0 和眼下的“大数据”，谁都想套套近乎。一些做理论的大佬也不能免俗，发明 RSA 算法的 R（Rivest）也带了几个学生转做神经网络学习问题的复杂性。一时间好不热闹。1993 年，美国电气电子工程师学会 IEEE 开始出版《神经网络会刊》，为该领域的高质量文章提供出版渠道。美国国防部、海军和能源部等也加大资助力度。神经网络一下子成了显学。

连接主义运动也培养了一堆新人，并使得加州大学圣地亚哥分校的认知科学系成为同类系科的佼佼者。鲁梅尔哈特后转往斯坦福大学任教，2011 年不幸死于已挣扎多年的神经退化疾病。乔丹（Michael Jordan）就是他的学生，而吴恩达（Andrew Ng）又是乔丹的学生。鲁梅尔哈特人虽离世，但香火没灭。他的另一名学生格鲁什科（Robert Glushko）后来远离本行，跟随硅谷互联网早期英雄塔南鲍姆（Marty Tennenbaum）创立了一家 XML 公司，那家公司后来卖给 Commerce One，赚了一票钱。格鲁什科捐钱设立了“鲁梅尔哈特奖”来奖励神经网络的研究者，辛顿成了第一位获奖者。麦克利兰德则先转往卡内基梅隆担任计算机和心理两系教授，后来也到斯坦福大学，在那里建立了“心、脑、计算研究中心”，还一度担任心理系主任。顺便说一句，塔南鲍姆的儿子约书亚·塔南鲍姆（Joshua Tennenbaum）现在都在麻省理工学院脑科学系当教授了。

辛顿则先转往卡内基梅隆大学，最终到加拿大多伦多大学计算机系任教。辛顿现在可是神经网络领域最牛的人了。他还有一段不太为外人所

知的革命家史：他是布尔的外曾曾孙子（对，就是“布尔代数”的那个布尔），他的曾祖母艾伦（Mary Ellen）是布尔的大女儿。中国革命的参与者、美国铁杆左派韩丁（William Hinton）和寒春（Joan Hinton）也是艾伦的孙子孙女。照这么说，韩丁是辛顿的堂叔，寒春是辛顿的堂姑。布尔的小女儿、艾伦的小妹妹伏尼契（Ethel Lilian Voynich）是传遍苏联和中国的小说《牛虻》的作者。《牛虻》西方不亮东方亮，在苏联和中国是几代人的革命加爱情励志畅销书。晚年在纽约生活陷入困顿的伏尼契，靠苏联和周恩来特批的稿费得以善终。这一家子把中国、苏联、革命、逻辑和神经网络都联系起来了，通吃“吃啥补啥”派和“想啥来啥”派。（智力题：伏尼契和辛顿是啥关系？）



韩丁（1919—2004）和
寒春（1921—2010）

语言学家、公共知识分子平克（Steve Pinker）对连接主义不以为然。鲁梅尔哈特和麦克利兰德在 PDP“圣经”中合作了一章，讲神经网络可以学会动词的过去式，比如一看 start，就知道 started，一看 come 就知道 came，等等。平克认为有规则的过去式（直接加 ed 的，如 started）可以通过简单计算得来；而不规则的（不通过加 ed 的，如 came）则是存在大脑的一个特定区域。平克引用神经心理学的证据指出，处理规则的和不规则的操

作是在大脑不同部位完成的，他还认为神经网络的行为和一类大脑受伤害患失语症的病人的行为相似。其实这种观察并不深刻，都是罗森布拉特 30 年前玩剩下的。符号系统可能比较适合处理规则的情况，而神经网络可能更适合不规则的情况，这个一般人都能想到。对神经网络派的批评也如此：我们可以定义一个规则，可以用符号系统实现，也可以用神经网络实现。哪个快用哪个。

符号处理和神经网络的方法论之争有时会被更大地夸张。伟大的乔姆斯基就不认可人工智能领域的最新进展。机器翻译历来是人工智能的试金石之一，就像在 1996 年之前的计算机下棋。机器翻译的早期实践都源于乔姆斯基的理论，但近来的突破却是基于统计的方法。乔姆斯基认为统计的方法不"优雅"（elegant），只是模仿而不是理解。会骑自行车不算理解，对自行车为什么不倒，能说清道理，才算理解。谷歌的研发总监诺维格（Peter Norvig）为统计方法辩护时说：简单的模型（如乔姆斯基理论，以及后来的各种改进版本）不能解决复杂的问题，人工智能的进一步发展必须两条腿走路。诺维格在加入谷歌之前曾是加州大学伯克利分校的计算机教授，他对两派都了如指掌，在学术界和工业界都被尊重，他写的《人工智能》是最流行的教科书。他的观点似乎被更多的人接受。

4. 深度学习

神经网络在 20 世纪 80 年代的光芒被后来的互联网掩盖了。但这几年，恰恰又是互联网产生的海量数据给了神经网络更大的机会。人工智能学者在计算机系曾经是最抬不起头的，这几年却人人都变成了公共知识分子。而人工智能领域最火的词儿就是"深度学习"。神经网络由一层一层的神经

元构成。层数越多，就越深，所谓深度学习就是用很多层神经元构成的神经网络达到机器学习的功能。

理论上说，如果一层网络是一个函数的话，多层网络就是多个函数的嵌套。网络越深，表达能力越强，但伴随而来的训练复杂性也急剧加大。目前对神经网络各种形态所对应的计算复杂性的研究并不多，从业者也以工程师、心理学家和统计学家为多。

辛顿是深度学习的先驱，他和学生在 2006 年发表的两篇文章开辟了这个新领域，其中登在《科学》上的那篇提出了降维和逐层预训练的方法，使得深度网络的实用化成为可能。深度神经网络最后几层的每个节点都可对应于某些概念。这是神经网络的一大进步，貌似为“吃啥补啥”找到了科学根据，调和了与“符号派”的矛盾。至于符号派买不买账，就是另一回事了。

深度学习的实测效果很好。辛顿一直用深度信任网络做图像识别，在 2012 年举办的图像识别国际大赛 ILSVRC（ImageNet Large Scale Visual Recognition Challenge）上，辛顿团队的 SuperVision 以绝对领先的成绩击败众竞争对手拔得头筹。该比赛用 1000 万张图像训练，用 15 万张图像测试，目标是识别测试图像到底是动物，是花儿，还是船，等等。在 2012 年前，普遍的错误率在 26%。但 SuperVision 头次参赛就把错误率控制在了 15%之下，以超过 10%的惊人优势遥遥领先。

2009 年，微软研究院的邓力小组开始和辛顿合作，用深度学习加上隐马尔科夫模型开发可实用的语音识别和同声翻译系统，2011 年取得突破。2012 年，微软负责研发的拉希德（Rick Rashid）在天津举行的一次会议上现场演示，他用英文演讲，机器用中文实时翻译，甚至中文合成的声音跟他自己的声色都非常相像。微软把这一成果迅速产品化，微软收购的

聊天工具 Skype 首先得益，整合了实时语音翻译的功能。此后，语音识别问题已经被认为彻底解决了。现在即使开源的软件都可以达到很高的识别率。中国的腾讯和科大讯飞等也都有此类产品。

年过 60 岁的辛顿不甘寂寞，和他的两个学生开了家专注深度学习的公司。公司成立没多长时间，谷歌和微软就对这家公司动了收购的念头，后来百度也加入竞标，最终花落谷歌，谷歌出了几千万美元，于 2013 年初收购了这家只有三名员工的公司。为了把辛顿纳入花名册，谷歌还真不差钱。

有意思的是，机器学习的几个主要研究团队都在加拿大，例如多伦多的辛顿、蒙特利尔的班乔（Yoshua Bengio）和阿尔伯塔的萨顿（Richard Sutton）。辛顿和萨顿的金主都是谷歌，他们陆续迁往加拿大都是在神经网络研究不招人待见的时期。无独有偶，王浩的学生库克（Steve Cook）也是在惨遭加州大学伯克利分校拒绝终身教职后迁往多伦多的。

2012 年，时任斯坦福大学人工智能实验室主任的吴恩达和谷歌合作建造了一个当时最大的神经网络，这是谷歌神秘的 X 实验室的一个计划。网络上一度疯传的谷歌猫脸识别就是用的这个参数多达十七亿的神经网络。后来，吴恩达在斯坦福大学又搞了个更大的神经网络，参数高达一百一十二亿。人脑的神经连接有一百万万亿个。从计算能力上说，如果这个人工神经网络要能接近大脑，每个人工神经元必须达到一万个大脑神经元的功能。这个神经网络会用到大量的图形处理芯片 GPU，GPU 一度是模拟神经网络的完美硬件，因为每个 GPU 芯片内都有大量的小核心。这和神经网络的大规模并行性天然相似。硬件的进步让以往不可能的成为了可能。GPU 的厂商 Nvidia 股票也一路飙升。对计算量的需求是没有止境的，新的芯片技术也被用到深度学习中，先是有人试图用 FPGA（可编程阵列）和 ASIC 实现各种深度学习算法，后来谷歌推出了专用芯片 TPU。



吴恩达（1976—　）

人工智能的统计派或神经网络派和逻辑派或符号派之争是从 1956 年达特茅斯会议开始的。明斯基的合作者佩珀特曾说神经派和符号派的区别就像分子生物学和进化生物学的区别，甚至有人因而争论大学的数学课应该以微积分为主还是以统计为主。新派自然是以统计为主。斯坦福大学人工智能实验室的创办人麦卡锡，是达特茅斯会议的主要组织者，"人工智能"这个词如果不是他最早提出的，至少是他最早使之流行的。也正是他把明斯基拉到了他当时任教的麻省理工学院。说他是"人工智能之父"是名副其实的，约翰大叔是铁杆的符号派。但后来的人工智能实验室主任却分别是做神经网络的吴恩达和李飞飞。这个转变也许是"吃啥补啥"派得志的风向标。斯坦福大学的这个神经网络的目标是模拟人的大脑。这让我们不禁想起了罗森布拉特，那不正是他的梦想吗？

参考文献指南

McCulloch and Pitts (1943)今天读来仍有意义。而 Rosenblatt (1958)和 Minsky and Papert (1969)则没必要花功夫去仔细研读了，只要知道结果就行了，就如想知道牛顿第二定律不一定非得读他的原著《自然哲学的数学原理》，

一本中学物理教科书足矣。Wang and Raj (2017)是详实的深度学习历史，但那是写给内行人看的。Hopefield 开创了神经网络研究的新气象，Hopfield (1982)讲离散型 Hopfield 网络，Hopfield and Tank (1985)讲连续型 Hopfield 网络。如果有理工背景，这两篇文章都不难懂，值得一看。深度学习的文献真不需要“指南”，各种深度的“深度”都烂大街了。

第 6 章
计算机下棋简史：机定胜人，人定胜天

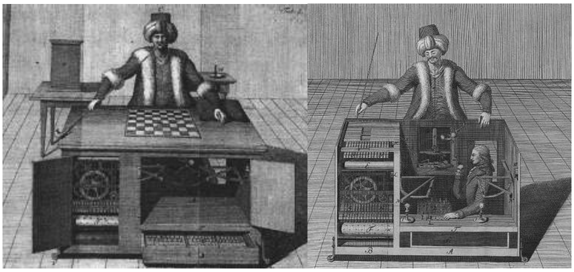
Play is the beginning of knowledge.
游戏是知识之源。
——George Dorsey（多尔西）

… because chess requires intelligence.
下棋需要智能。
——Alan Turing（图灵）

1. 机器下棋史前史

1769 年，德国发明家兼外交家肯佩伦（Wolfgang von Kempelen）男爵准备造一台机械的下棋装置，一年后机器完工，取名“土耳其人”（The Turk），那时大家就把这玩意叫作“自动机”（automaton）。肯佩伦把这台机器展示给奥匈帝国的掌权者特蕾西娅（Maria Theresia，奥国女大公、匈国女王），于是它就成为娱乐欧洲各皇室的保留节目。称为“土耳其人”是因为这个装置的后面坐着一个土耳其装束的木头人。1804 年，男爵死后，“土耳其人”被转卖给德国发明家兼娱乐人马泽尔（Johann Nepomuk Maelzel），1809 年马泽尔把它展示给拿破仑，并和这位欧洲不可一世的征服者对弈一局。拿破仑执白棋先手，但最后“土耳其人”大胜，拿破仑恼羞成怒，把棋盘

上的棋子全胡撸到地上。有好事者把拿破仑和“土耳其人”的对战棋谱记录在案，确实艺不如“机”。陆续和“土耳其人”接触过的名人还有富兰克林、爱伦坡和数学家巴贝奇。


“土耳其人”

“土耳其人”在欧洲巡演了几十年，最后被人发现是个彻头彻尾的假货：那个装置里总是有个活人，而且是个下棋高手。肯佩伦只是发明了个魔术而已。那时的水平，想造台会下棋的机器，门儿都没有。1827 年，“土耳其人”到美国巡演时，请了美国当时的顶级高手施伦伯杰（Schlumberger）藏匿其中。在巴尔的摩的一次表演中，两个孩子发现施伦伯杰频繁出入后台，发现了这个秘密并透露给报界。见过这台机器的高人（如富兰克林和巴贝奇）一开始就猜这是魔术而不是科技。但当时还是有很多人愿意相信“土耳其人”真会下棋。1838 年马泽尔和施伦伯杰相继去世，随后“土耳其人”也退役，被费城的中国博物馆收藏，1854 年博物馆的一场大火彻底摧毁了“土耳其人”。

和牛顿、霍金一样，巴贝奇还做过一届剑桥大学的卢卡斯教授，他对所有机器装置都感兴趣，他在看到“土耳其人”时，正在研制第一台机械计算机“分析机”（analytical engine）。他认为他的分析机也可以下棋，但那至多是猜测。

下棋一直就是人类智能的挑战，自然也成了人工智能的标志之一。

二战没结束时，图灵就研究计算机下棋，他 1947 年编了第一个下棋程序，可惜那时计算机的时间（简称“机时”）很宝贵，轮不到他上机——地主家没余粮，图灵也不能保证机时。但即使后来拿到机时，那机器和程序的水平也很有限。米奇（Donald Michie）是图灵的追随者，1950 年试着在纸上模拟程序，和图灵对弈，但这实在不是办法。图灵在曼彻斯特大学的同事普林茨（Dietrich Prinz）接着图灵的思路，在 1951 年写了一个残局程序，能在离将死还有两步的情况下，找到最优解。这个问题也被称为“两步将死”（mate-in-two）问题。

2. 跳棋插曲

1951 年，图灵的朋友斯特拉切（Christopher Strachey）在曼彻斯特 Mark-1 上写了第一款跳棋程序。图灵在 1952 年曾与之对弈一局，轻松取胜。1956 年 IBM 的塞缪尔（Arthur Samuel，人工智能里程碑达特茅斯会议的参加者之一）写了第二个跳棋程序，这款程序的特点是自学习，这也是最早的机器学习程序之一，后来不断改进，曾经赢过盲人跳棋大师。

20 世纪 80 年代末，最强的跳棋程序一直就是加拿大阿尔伯塔大学的 Chinook，作者是现任阿尔伯塔大学理学院院长的计算机系教授舍佛（Jonathan Schaeffer）。数学家丁斯利（Marion Tinsley）自 20 世纪 50 年代起就一直是跳棋的人类冠军。丁斯利对跳棋理论研究很深，对舍佛团队也很支持，但美国、英国和加拿大的跳棋协会一直拒绝 Chinook 参赛。为了和 Chinook 比赛，丁斯利放弃了他的冠军称号。1992 年丁斯利大战 Chinook 并取胜，1994 年再战，但在比赛期间不幸确诊患了胰腺癌，不久病逝。丁斯利的公开纪录，除了输给 Chinook 几局棋外，从没有输给过任何人类

棋手。此后 Chinook 孤独求败。

舍佛团队继续精研跳棋理论和实践，直到 2007 年，他们证明对于跳棋，只要对弈双方不犯错，最终都是和棋，而 Chinook 已经可以不犯错。他们的结果发表在 2007 年 9 月的《科学》杂志上，自此跳棋这一页就算翻过去了。舍佛的兴趣遂转向德州扑克和围棋。

3. 计算机下棋之初

几乎和图灵同时，冯诺伊曼也在研究计算机下棋，他和经济学家摩根斯顿合作的《博弈论》1944 年出版，其中首先提出两人对弈的 Minimax 算法。香农（Claude Shannon）1950 年在《哲学杂志》发表“计算机下棋程序”（Programming a Computer for Playing Chess）一文，开启了计算机下棋的理论研究，其中主要思路在“深蓝”和 AlphaGo 中还能看到。有趣的是，战时图灵在布莱彻利庄园破解德国密码，而香农在贝尔实验室研究密码理论，其中还用到了他后来发明的信息论。图灵的工作涉及军事，直到 1974 年才部分解密，而香农则偏理论。图灵战时到访美国普林斯顿大学和贝尔实验室，曾和香农多次会晤，但他们从来没聊过密码学，尽管香农猜到了图灵在干啥。1950 年香农去英国参加信息论会议时到曼彻斯特大学图灵的办公室回访，他们这次只聊了下棋和大脑，仍然没聊密码。图灵没有参加这次在伦敦的会议，但贡献了两篇短文，一篇讲机器学习，另一篇讲下棋。直到图灵的工作解密，香农才知道图灵在战时已经用到了“熵”，但是不是从了解信息论的美国同事处学来的就无从考证了。信息安全专家史密斯（S. W. Smith）曾写过一篇题为“图灵来自火星，香农来自金星”的文章，很明显这是受那本《男人来自火星，女人来自金星》的启发。

香农把棋盘定义为二维数组，每个棋子都有一个对应的子程序计算棋子所有可能的走法，最后有个评估函数（evaluation function）。传统的棋局都把下棋过程分为三个阶段：开局、中局和残局，不同阶段需要不同的技术手段。香农的论文引用了冯诺伊曼的《博弈论》和维纳的《控制论》。

Minimax 算法中，二人对弈的一方为 max，另一方为 min，max 一方的评估函数要越高越好，min 一方的则越低越好。max 和 min 的对弈就形成了博弈树。树的增长是指数式的，当树很深时，树的规模会变得不可控。达特茅斯会议的组织者之一麦卡锡首先提出 α-β 剪枝术以控制树的增长。纽厄尔、司马贺和肖（Newell，Simon，Shaw，简称 NSS）在他们著名的定理证明程序之后，又做了下棋程序。他们首先实现了 α-β 剪枝术，其程序是在一台 Johnniac 上实现的。原始的 Minimax 算法是在博弈树被全部画出后再静态地计算评估函数，而 α-β 剪枝术则采取边画树边计算评估函数的动态方法。当评估函数的值超越给定的上界和下界时，树的搜索过程就停止，这样大大减少了树的规模。平均而言，在同样资源限制下，α-β 剪枝术要比原始 Minimax 算法搜索的树深度多一倍，也就是说，可以比 Minimax 向前看的步数多一倍。

第一个可以走完全局的下棋程序是 IBM 的工程师伯恩斯坦（Alex Bernstein）1958 年在一台 IBM 704 上做的。估计那时 IBM 支持下棋就像后来支持“深蓝”和谷歌支持 AlphaGo 一样，虽没什么短期实用价值，但是很好的公关。机器每步要花 8 分钟想，随便会走几步棋的人就能击败这个程序。

1959 年，麻省理工学院的几位本科生在当时刚到校任教的麦卡锡指导下开始学习计算机下棋，他们 1962 年本科毕业时，用 Fortran 实现了一

款实战下棋程序，跑在 IBM 新出的 7090 大型机上，此时已经可以击败一般的象棋初学者了。这个结果变成了其中一位学生 Kotok 的本科学位论文。1962 年麦卡锡前往斯坦福大学任教，他进行了持续改进，这个程序后来被称为 Kotok-McCarthy 程序。

1966 年，美苏的对抗也扩展到计算机下棋。苏联科学院的理论与实验物理研究所（ITEP）也在本所研制的一台 M20 计算机上开发了一款下棋程序，他们要和斯坦福大学的 Kotok-McCarthy 程序一决高下。从 1966 年 11 月 22 日开始，直到 1967 年 3 月 10 日止，他们通过电报的方式走了四局。最后苏联 3:1 战胜美国。

当时麻省理工学院的程序员格林布拉特（Richard Greenblatt）也在改进 Kotok 的程序，还是位成绩不错的棋手。他在 PDP-6 上实现了程序 MacHack VI。1966 年，一直批评人工智能的哲学家德雷弗斯[①]也和 MacHack 对弈过一局，并且输给了 MacHack，这倒没有改变他对待人工智能的态度。1967 年 MacHack 参加象棋锦标赛，并累计积分 1400 ，这相当于不错的高中生水平。这个程序用了 16KB 内存，后来 PDP 的厂家 DEC 把它预装到所有 PDP 系列的机器中。MacHack 也是 Internet 前身 ARPANET 上最早的网络游戏。当时给格林布拉特帮忙的志愿者中有个人叫克柔可（Steve Crocker），他后来成为 Internet 前身 ARPANET 的重要人物，并创办了互联网标准化组织 IETF 且写了第一个标准化文本 RFC。

司马贺在 1957 年预言十年内计算机下棋程序可以击败人类，明显未果，于是他在 1965 年再度预言这个目标在 20 年内可以实现。1968 年国际象棋大师列维（David Levy）和麦卡锡打赌十年内机器不可能赢列维。1978 年最厉害的计算机程序 CHESS 和列维比了一盘，自然是列维赢，麦卡锡

① 那时德雷弗斯的那本后来引起争议的《计算机不能干什么》还没出版。

为此输了 500 英镑。CHESS 在 20 世纪 70 年代末 80 年代初一直是计算机下棋的冠军。此时尚看不到计算机短期内可以赢人的可能性。

1971 年，当年击败 Kotok-McCarthy 的苏联小组改进了他们的程序，新程序名叫 KAISSA（象棋女神）。KAISSA 和传奇大师斯帕斯基（Boris Spassky）赛了两局，一负一和，这个战绩惊动了棋界。1972 年 KAISSA 接受了《共青团真理报》的挑战：KAISSA 将和读者下两盘，《共青团真理报》从读者寄来的各种走法中挑出推荐最多的。这其实就是“众包”概念的雏形。最终，KAISSA 还是一负一和。但 1972 年斯帕斯基却又输给美国怪人菲舍尔（Bobby Fischer），这是美国第一次在国际象棋领域战胜苏联，尼克松稍晚在会见勃列日涅夫时送了对手一副袖珍国际象棋，成心恶心人家。

1970 年开始，美国计算机学会（ACM）的年会都在晚餐时举行计算机象棋比赛，作为娱乐节目。CHESS 连着四年都是冠军。第二届时，纽厄尔的学生柏林纳（Hans Berliner）参加了，取得第二名，这鼓舞了纽厄尔，他决定把柏林纳留校，专职在卡内基梅隆大学研究计算机做计算机象棋。柏林纳本人也是国际象棋高手，曾赢得美国通讯赛冠军，他留校后，并没有走教学制（tenure track），而是走了研究口——卡内基梅隆的研究制教员也分三级，研究员（Research Scientist）对应助理教授，高级研究员（Senior Research Scientist）对应副教授，而首席研究员（Principal Research Scientist）对应正教授。后面几年的比赛，都有纽厄尔的学生参加，成绩不错。

1974 年，为了给在瑞典斯德哥尔摩召开的“国际信息联合会大会”（IFIP）找点乐子，组织者仿效美国计算机学会年会的做法，举行了第一届世界计算机象棋锦标赛。锦标赛的组织者是刚从哥伦比亚大学转往加拿大麦吉尔大学的牛伯恩（Monty Newborn）。除了计算机下棋，牛伯恩的另

一个兴趣是机器定理证明，他写过两款定理证明程序，参加各种定理证明比赛。尽管他的下棋程序和定理证明程序在比赛中并没有出色表现，但他写的下棋和定理证明的书却很有意思。第一次锦标赛，除了美国和加拿大的几位高手外，还邀请了欧洲的几个团队，当然要包括苏联神秘的 KAISSA。KAISSA 击败了在 ACM 年会拿了四次冠军的 CHESS，赢得头筹，报了两年前斯帕斯基输给菲舍尔的一箭之仇。勃列日涅夫倒是没把当年那副袖珍棋送还给已经下台的尼克松，不知算是小气还是大度。

进入 20 世纪 80 年代，又出了新一茬象棋程序。当时最厉害的两个电脑棋手，一个是跑在超级计算机克雷上的 Blitz，另一个则是贝尔实验室的专用机器 Belle。Belle 的发明人之一汤普森（Ken Thompson）那可是了不起的人物，在计算机界无人不晓。他最杰出的成绩是发明了 UNIX 操作系统（现在苹果操作系统、波音 747 的飞行系统和安卓手机操作系统都是 Unix 的变种）和 C 语言，1999 年被克林顿授予美国“国家技术奖章”。他在计算机下棋上的贡献多少被略视了。在 1982 年的北美计算机象棋锦标赛上，Belle 击败了 Blitz。Belle 是第一个取得“大师”称号的计算机棋手。1982 年在去苏联比赛的路上，Belle 被美国政府在肯尼迪机场海关没收，理由是企图向苏联输送先进武器，Belle 的终端里确实有当时对苏联禁运的超大规模集成电路。拖了小一年功夫，最后汤普森等破费了 600 美元罚款，才赎回 Belle，但比赛耽误了。

20 世纪 80 年代，机器之间的比赛此起彼伏，但机器和人之间仍然有着不可逾越的鸿沟。1980 年，天才弗雷德金（Edward Fredkin）专为计算机下棋建立了弗雷德金奖金，奖有三等，头等奖是 10 万美元，奖给第一款能赢现任世界冠军的下棋机器。

4. “深蓝”

20 世纪 80 年代中期，卡内基梅隆大学的柏林纳开始用专用硬件来实现下棋机，他的成果 HiTech 马上成为最强的机器棋手。这时来自中国台湾的许峰雄到卡内基梅隆大学计算机系跟随孔祥重读计算机体系结构（常说的“硬件”）方向的博士，孔祥重是孔子的嫡孙。许峰雄的室友很快把他拉到 HiTech 项目帮忙设计一个硬件的评估函数，但许峰雄却和柏林纳关系不睦。在资金有限的情况下，许峰雄和几个研究生利用业余时间快速开发出了 ChipTest，而 ChipTest 立即变成了 HiTech 的竞争对手，并受到柏林纳的打压。许峰雄在计算机系也变成众矢之的，每次都是靠导师孔祥重的帮忙化险为夷。ChipTest 的改进版“深思”（Deep Thought）1989 年赢得弗雷德金二等奖：成为第一个国际象棋特级大师的机器棋手，累计分超过 2400。随后 HiTech 也加入这个行列。而此时 IBM 意识到“深思”的商业价值，于是劝说整个团队在毕业后加入 IBM，开发下棋机，把对手锁定为当时的世界冠军俄罗斯特级大师卡斯帕罗夫。卡斯帕罗夫对机器下棋非常熟悉，他在屡次和机器对决后曾说：机器下棋没有洞见（insight）。

IBM 的外号叫 Big Blue，于是新的项目 1996 年被命名为“深蓝”（Deep Blue）。1996 年 ACM 年会的闭幕节目是“深蓝”对决卡斯帕罗夫，六局棋。“深蓝”旗开得胜，第一局就赢了老卡，最后还是老卡 4:2 赢得决赛。但此时老卡对“深蓝”刮目相看，他说机器对手不光有洞见，而且有几步简直像“上帝下的”。

第二年“深蓝”和老卡再战，老卡号称要捍卫人类的智力尊严。他赢了第一局，但随后则越来越保守，彻底输给“深蓝”。老卡下棋过程非常情绪化，这有时会给人类对手施加压力，但“深蓝”压根不吃这套。

1997 年 5 月 11 日，老卡认输，“深蓝”成了第一位战胜当时世界冠军的机器。事后，卡斯帕罗夫回忆：第二局是关键，机器表现超出他的想象，它经常放弃短期利益，表现出非常拟人的危险（“showing a very human sense of danger”）。

在“深蓝”赢了卡斯帕罗夫之后，职业棋手并没有因此而改行，他们反而更多地依赖计算机来训练。而职业比赛的解说者也越来越多地借助计算机程序来分析解说一场比赛。机器作为教练，反而更快地帮助人类棋手进步。有美国高中的象棋教练观察到从来就没有过这么多年轻棋手在年龄很小时就积分这么高，这都得益于计算机教练，因为过去的孩子从来就没有机会能和特级高手比赛。

瑞典青年棋手卡尔森（Magnus Carlsen）就是如此。内行说卡尔森的下法很像计算机。2013 年卡尔森在印度的金奈（Chennai）客场击败印度老将、卫冕冠军阿南德。现在两台个人电脑下棋，人已经看不懂它们在下什么。尽管如此，“深蓝”队员，同样毕业于卡内基梅隆大学的坎普尔（Murray Campbell）仍然不认为机器有智能。这其实是整个“深蓝”团队的意见，他们都不是人工智能出身，反而和同系的人工智能教授结下梁子。“深蓝”获胜后，美国人工智能学会（AAAI）曾经组织过一个研讨会，对人工智能启发式搜索做出过杰出贡献的加州大学洛杉矶分校教授科夫（Rich Korf）曾不满“深蓝”团队的立场。

5. 围棋和 AlphaGo

相对于计算机在国际象棋中的胜利，中国象棋的进展一直落后。一些外行认为中国象棋要比国际象棋难，其实非也。“深蓝”胜利之后，聪

明人认为计算机下棋这事已经到头了，没人愿意费力不讨好，IBM 也解散了“深蓝”团队。迟至十年后的 2006 年，中国象棋程序开始击败特级大师级别的人类棋手。倒是围棋确实更具挑战性，但围棋在西方没什么受众，自然没什么聪明人愿意花时间。

围棋的棋子多，组合可能性也多，画出博弈树的所有可能枝叶后，在上面跑 α-β 不太经济。于是聪明人想到了蒙特卡洛方法。蒙特卡洛方法最常用的教学例子就是计算圆的面积：在一个正方形里贴边画一个圆，然后随机向这个正方形里扔沙粒，扔到足够多时，开始数有多少沙粒落在圆里，结果除以所扔沙粒总数再乘以正方形面积，就是圆的面积。

思路和求圆的面积类似，随机模拟对弈双方走棋。当走棋的次数很多时，就可算出下棋点的概率，然后挑概率最大的地方落子。谷歌的 AlphaGo 首次引用了强化学习（Reinforcement Learning），使得机器和自己对弈学习。强化学习的发明者是巴托（Andy Barto）和他的学生萨顿（Richard Sutton）。说来话长，冯诺伊曼在普林斯顿大学设计计算机的主要助手是伯克斯（Arthur Burks），伯克斯培养了有史以来第一个计算机科学的博士霍兰德（John Holland）。霍兰德发明了遗传算法。霍兰德的一个有名的学生是科德（Edgar Codd），因发明关系数据库而获图灵奖。巴托是霍兰德的一个大器晚成的弟子，巴托博士毕业后被神人阿比卜（Michael Arbib）招到麻省大学计算机系，在那里培养了自己的第一个博士生萨顿。在经历了人工智能的此起彼伏后，萨顿现在阿尔伯特大学任教。正是他和已经转向围棋研究的舍佛的碰撞，萌生了把强化学习应用到围棋的想法。在神经网络的低潮期，巴托和萨顿利用他们所在的麻省大学计算机系和 GTE 实验室资助了一帮同仁，其中就有后来成大器的麦克·乔丹。

强化学习从 20 世纪 80 年代就被发明，但一直不被重视，是 AlphaGo

使得它发出亮光。萨顿还值壮年，AlphaGo 团队里就有 4 个萨顿的学生，其中包括首席科学家席尔瓦（David Silver）。巴托老兵不死，在做了一届计算机系主任后，几年前从麻省大学退休了。退休前，他终于看到强化学习渐成显学，他和萨顿合著的《强化学习》马上要出第二版了。

参考文献指南

Standage (2002)是关于最早的下棋机“土耳其人”最完整的描述，有兴趣深究历史的读者不妨翻翻这本书。

牛伯恩（Monty Newborn）是加拿大麦基尔大学计算机系的教授，他的主要研究方向一是机器定理证明，二是计算机下棋，而且他是实战派，他和他的学生做过几个机器定理证明的程序和计算机下棋程序，尽管成绩都不是很好，但他为学界做出了贡献。Newborn (1975)对机器下棋的早期历史有详实考证。

许峰雄的《深蓝解密》（Hsu (2004)）是一本关于 IBM“深蓝”历史的书，对卡内基梅隆大学计算机系的内部政治也有精彩披露，可读性很强，近乎历史小说。而 Newborn (2002)是一本关于 IBM“深蓝”更加偏重技术也更加公正的读物。

第 7 章

自然语言处理

The noblest pleasure is the joy of understanding.

最高级的快乐就是理解的愉悦。

——Leonardo da Vinci（达芬奇）

It is not our aim to refine or complete the system of rules for the use of our words in unrelated of ways.

完善规则系统用于无关的字词，绝不是我们的目的。

——Ludwig Wittgenstein（维特根斯坦）

1. 乔治敦实验

1953 年至 1954 年，IBM 资助美国乔治敦大学（Georgetown）进行了有史以来的第一次机器翻译。早期的几家计算机公司之间因为知识产权混战，各败俱伤，IBM 异军突起，成了新兴的计算机产业领头羊。作为公关兼广告，IBM 顺势扶持了几个项目，其中有计算机下棋的，也有机器定理证明的。那时“人工智能”这个词儿还没被发明。机器翻译自然也是一个被关注的领域。IBM 此次出场的是 1953 年刚刚发布的 701 计算机。

乔治敦实验的目标很明确：把几十个俄文句子翻译为英文，总共定义了 6 条语法规则，250 个单词，主要领域是有机化学。那时还没有交互

式输入，所有要翻译的句子都要事先凿到卡片上。在美国制造的计算机上输入俄文还不方便，只能先把俄文字母用对应的罗马字母编码。20 世纪 60 年代初期，IBM 还试图把类似的系统用到中英文翻译，搜集情报是优先的军事应用。

乔治敦实验要早于达特茅斯会议，这也从另一个角度证明了达特茅斯会议更多是“人工智能”这个词儿流行的源头，而不是这门学科的起点。乔治敦实验为计算机的应用烧了把火，自动翻译成为多人扎堆的研究课题。美国空军和美国原子能委员会（Atomic Energy Authority），以及欧洲原子能发展组织（Euratom）都安装了 IBM 的翻译系统，企图解决翻译苏联文献的问题。

到了十年后的 1964 年，美国政府的科研资助机构意识到机器翻译的研发进展缓慢，于是责成美国科学院对现状做一总结。美国科学院遂成立自动语言处理顾问委员会（Automatic Language Processing Advisory Committee，简称 ALPAC）。经过两年调研，1966 年 11 月，ALPAC 发布了《语言与机器》报告，称机器翻译比人翻译要慢，更不准确，而且成本更高（估算比人要贵两倍）。结论是机器翻译在可预见的未来没法实用，应该立即停止对机器翻译的资助，转而支持一些更基础的、辅助性的研究，如电子字典等。这个报告第一次提出了“计算语言学”的概念。

2. 乔姆斯基和句法分析

乔姆斯基之于语言学和认知科学，就像图灵之于计算机科学。没有这些“先知”，我们不知还要在黑暗中摸索多久。这个说法不只是比喻性的：乔姆斯基的句法频谱后来被证明和几种自动机有着深刻的关联：乔姆

斯基 3 型文法（正则表达式）等价于有限自动机，2 型文法（上下文无关文法）等价于下压自动机，1 型文法（上下文相关文法）等价于线性有界非确定图灵机，0 型文法等价于图灵机。乔姆斯基的句法研究导致了乔姆斯基在哲学上的理性主义立场，这与英美的经验主义主流不合拍。其实乔姆斯基的理性主义与欧陆传统的理性主义并不完全一致，倒是和丘奇–图灵论题可互为佐证。这话展开了讲太长，语言学和哲学系找不到课题的博士生可以试试这个。

乔姆斯基决定干语言学其实缘于他的政治兴趣。他是结构主义语言学开山哈里斯（Zellig Harris）的学生，他们都是犹太人，有同样的政治主张。大二时，乔姆斯基对学业困惑，准备退学，哈里斯劝他说：你干嘛不试试语言学呢，可先从数学和哲学入手。哈里斯给了他一本自己尚未出版的《结构语言学方法》一书的草稿，乔姆斯基从此开了窍，走上了语言学之路。传统的人文范儿的语言学家必是那些懂多种语言的人才。但哈里斯把语言学从人文转变成科学，他在宾夕法尼亚大学（UPenn）建立了美国第一个语言学系。乔姆斯基在宾大得了本科和硕士学位后，在那时还在宾大哲学系教书的古德曼（Nelson Goodman）的影响下，前往哈佛投奔当时美国哲学界的领袖蒯因（Quine），他在哈佛还被选为初级研究员（Fellow）。哈佛的这个 Fellow 是给那些明日学术之星准备的，在乔姆斯基之前，王浩、库恩等都得过。

在哈佛期间，乔姆斯基发表了他的第一篇学术论文“句法分析系统”（Systems of Syntactic Analysis）。值得一提的是，这篇文章并未发表在语言学杂志上，而是在数理逻辑最权威的《符号逻辑杂志》（*JSL*）上。这本杂志由丘奇创办，从杂志创刊开始，丘奇就为 *JSL* 写评论，一直写到他 80 岁高龄。公正地讲，丘奇和 *JSL* 就是现代逻辑史。丘奇 1937 年为图灵那篇惊

天之作写评论时把图灵的装置命名为“图灵机”。后来 *JSL* 的主编克里尼（丘奇的学生）也曾漫不经心地拒绝了纽厄尔、司马贺和肖的“逻辑理论家”的文章，由此引发纽厄尔和司马贺对逻辑学家和定理证明方法终身不满，成为人工智能各种派系之争的一种。



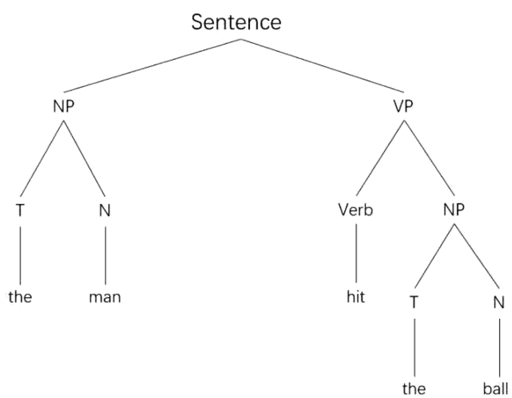
乔姆斯基（1928— ）

1955 年 4 月，还在哈佛的乔姆斯基收到了征兵通知，按照当时的惯例，如果得到博士学位就可以免服兵役，但他在短期内无法满足哈佛的博士学位要求。于是乔姆斯基又回到宾大找到哈里斯和古德曼，他们为乔姆斯基开了后门。乔姆斯基只是书面回答了几个问题，6 个星期后，他就从宾大拿到了博士学位。这比维特根斯坦的剑桥博士还要“速成”，维特根斯坦当时也是靠罗素、摩尔、凯恩斯和拉姆齐等一票剑桥大佬的提携。

乔姆斯基的博士论文来自他当时正在写的一本近一千页的大部头《语言学理论的逻辑结构》（*The Logical Structure of Linguistic Theory*）。这本书 20 年后才正式出版，书中包含的内容成为后来乔姆斯基句法分析的雏形。最早认识到乔姆斯基重要性的是哈佛年轻的教授、认知心理学家米勒（George Miller），据说他是《语言学理论的逻辑结构》的第一个读者。米

勒后来在普林斯顿大学还领导了“词网”（WordNet）的工作，这算是最早的知识图谱。

按照乔姆斯基句法分析，句子可以通过一系列规则得到解析。一个句子可以解析成名词词组（NP）和动词词组（VP），而名词词组和动词词组又可再被解析。例如句子 The man hit the ball（“这个人打了那个球”）就可解析为如下的树。



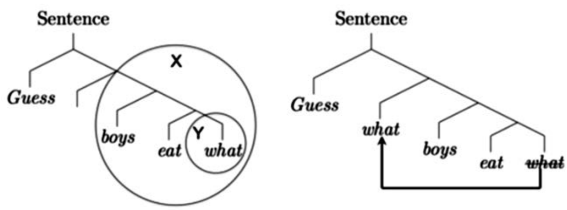
句法解析树

乔姆斯基认为，所有的语言（人工或自然）都有与此类似的句法结构，并进一步指出语言的结构是内在的，而不是通过经验习得的（acquired）。乔姆斯基的理论在《句法结构》后，几乎每十年就来一次变革。20 世纪 70 年代有扩展标准理论，80 年代又有管约（Government Binding），90 年代又有最简原理。

尽管乔姆斯基的语言观点不断变化、演进，但有些基本的东西始终未变。他在 2014 年还在和他的同事 Berwick 合作学术论文[①]，企图证明语

① 乔姆斯基 2014 年还和他的朋友及同事 Ian Tattersall 和 Robert Berwick 合作发表学术论文，见 Bolhuis 2014。2016 年他和 Berwick 合作的书 *Why Only Us* 出版。

言机制的核心是简单的“合并”（merge）操作，merge 是人类通过进化而特有的能力。两个句法单位 a 和 b，经过“合并”就成了集合{a, b}。例如句子 boys eat apples，就是{boys, {eat, apples}}。下图是句子 Guess what boys eat 的解析树。X 对应于子树{boys, {eat, what}}，Y 是 X 的元素 what。如果把 X 和 Y 合并就成了{what, {boys, {eat, what}}}。在这里，第一个 what 就成了量词，于是句子可以表示为“for what *x*, boys eat *x*.”，或者，用一阶逻辑的符号就是（all *x*）Eat（boys, *x*）。所谓“合并”不就是 LISP 中的最基本算子 cons 嘛。



“合并”是语言的基本操作

哲学传统素有英美经验主义和欧陆理性主义之分。经验主义靠近科学，理性主义靠近数学。但进入后现代的欧陆哲学，已经成为所有科学人所不齿的玩意儿，这是后话。乔姆斯基出名后，马上就批评邻校哈佛的行为主义心理学家斯金纳（Burrhus Frederic Skinner）。从某种意义上说，行为主义是极端的经验主义。所有黑箱子理论，无论是神经网络还是统计派，在乔姆斯基眼里都属行为主义。乔姆斯基认为，理论应该先于事实，这明显违背“实践是检验真理唯一标准”的说法。他常用遗传学祖师爷孟德尔为例，但孟德尔常常删改不支持理论的数据。

1990 年《科学美国人》的记者霍根（John Horgan）采访乔姆斯基时机智地问他：你在政治上一直是反体制的（anti-establishment），但你在学术上本人就是体制。老乔面无表情地说：“在语言学上我一直是少数派。”这

听起来来有点像玩笑，但乔姆斯基很严肃：欧陆的理性主义在英美确属少数派。这倒是提供了一种解读乔姆斯基的新角度。很多哲学家对乔姆斯基持理性主义立场不解，但对于计算机科学家而言，这却很自然。乔姆斯基文法和各种自动机的关系是计算机科学的理论基石之一。计算机科学也许是经验和理性的桥梁。

人是否有智能，就像说人是否有人性，实在不好回答。行为主义的图灵测试也涉及语言。乔姆斯基则提出，人对语言的创造性使用能力（creative aspects of language use）是人性的标志。这可以看作图灵测试的一个变种。乔姆斯基认为机器是被迫行动（compelled to act），而人则是主动行动或煽动（incited inclined to act）。compelled 和 incited 之区别可能就是牛津日常语言学派所谓的“意向”。人比机器强在于人的创造性。笛卡儿所说的广义和哲学语法（general and philosophical grammar）就是思维的基本法则，语言是心灵的镜子。乔姆斯基也主张语言是一种社会现象，是共同体的共享资产（shared property of community），这同早期美国实用主义立场接近，“共同体”概念是实用主义的核心。乔姆斯基也自认他是实用主义鼻祖皮尔士在语言学的传人。在乔姆斯基之前，语言学也企图把自己当作科学，但那只是人为的分类，就像早期的生物学，直到有了乔姆斯基的句法理论，语言学才有了坚实的基础，就像生物学有了化学和分子生物学的理论和工具。乔姆斯基认为心身（mind-body）问题是个伪问题，难度倒不在于如何定义 mind，而在于连什么是 body 这样貌似简单的问题都无法明确地说清。他认为 mind 的研究终究会变成像物理学、化学那样的学问。只不过现在还要用心理学的术语逐步获得进展。语言学是突破口之一，由此可以找到“心”的物理机制。从这个意义上说，乔姆斯基也不完全反对经验主义。

乔姆斯基比较了笛卡儿和牛顿的理论，认为牛顿为物质世界提供了一个解释理论，但笛卡儿却没有为语言的创造性使用提供满意的解释。他自认为他正在向这个方向前进。乔姆斯基批评皮亚杰的工作不严谨，没有基本的形式化的精准（formal precision），研究心理学时，文学有时比科学更管用。在乔姆斯基眼里，科学史大概比科学哲学更有用。他把认知科学和神经生物学的关系比作化学和物理的关系，只不过认知科学和神经生物学类似伽利略之前的物理学，还在探索。也有人称乔姆斯基是语言学的牛顿。科学方法素有解释（explanation）和描述（redescription）之分。统计方法可看作一种描述，但不是解释。乔姆斯基不认可语言学的统计方法。

乔姆斯基是活着的人里被引用次数最多的知识分子，即使从苏格拉底算起，他的引用数也可排进前十。他的时事评论几十年来都被广为关注，这一点颇像他的偶像罗素。乔姆斯基的独特政治观点体现在他对当代政治事件的评论上。人们轻率地把乔姆斯基划为左派，其实，他是反建制者，永远怀疑权威，永远同情人民。乔姆斯基作为犹太人，却不被以色列接受，因为他同情巴勒斯坦的立场。以色列甚至拒绝给乔姆斯基发签证。乔姆斯基在任何地方的学术演讲，最后总要“饶”一段儿同等时间的政治评论，就像演出的返场。

乔姆斯基敬仰的人不多，无政府主义者乔治·奥威尔是一个，罗素是另一个。乔姆斯基办公室内就挂着一幅罗素的照片。很多人拿乔姆斯基和罗素做比较，罗素在出版了《数学原理》后很少再有原创的知识贡献，兴趣转向政治；而乔姆斯基在《句法结构》之后也成为一位社会活动家和公共知识分子。但乔姆斯基仍然不断有科学成果出来。罗素被下过两次大牢，乔姆斯基 1967 年因为反越战被捕，和诺曼·梅勒关在一起。

乔姆斯基同辈的人中参与政治的不少，王浩、普特南也都是活跃分

子，但他们都没有乔姆斯基那么坚定与持久。他们的年龄很尴尬，上面有蒯因那样的学术霸主，下面又有更激进的年轻辈的学生，学生一般不会带他们玩。但在经历了几代政治思潮之后，乔姆斯基在 80 多岁高龄时，仍然被年轻人认为是自己人。每次看到乔老爷那张平静异常的脸，我都会叩问良心。

麻省理工学院为了保护他们的“院宝”，屡次为乔姆斯基雇保镖。尽管乔姆斯基本人反对，但他确实时时处于危境中，他的名字也出现在著名的邮件炸弹（Unabomber）卡津斯基（Kaczynski）的黑名单上。梅勒回忆和乔姆斯基的狱中一夜时说：乔姆斯基真是个敬业的老师，一想到周一他不能给学生们上课，就面露不安。如果我们听罗素的音频，或看乔姆斯基的视频，丝毫感觉不到在阅读他们的作品时伴随的兴奋和快感。鸡蛋确实比鸡肉好吃。

乔姆斯基反政府，但他成名的文章“三种模型”确实是陆军和海军的研究经费支持的。他当时所在的单位之一现代语言学系后来演变成语言与哲学系，所在的另一单位电子学研究所 RLE（Research Lab of Electronics）从战时的放射实验室演变而来。当然，做人工智能的没有政府资助几无可能。NSF 的资助规模有限，DARPA、海军和空军是最大的资助者。斯坦福大学的维诺格拉德是很少的例外，他从来就没写过 DARPA 项目计划书，除了他到斯坦福大学的第一年受泽于麦卡锡的 DARPA 项目外，他后来的研究主要是施乐实验室（XEROX PARC）资助的。

3. ELIZA 和 PARRY

魏森鲍姆（Joseph Weizenbaum）1923 年生于柏林，父母是德国犹太人，

因惧怕德国的恶劣环境，1936 年全家迁往美国。魏森鲍姆一辈子讲话都带着德国口音，他晚年的许多著作都是德文，尽管他在韦恩州立大学读书时还曾参加过美国空军。1960 年，他得了数学硕士学位后就进了工业界，在通用电气的工作经历把他引到了刚刚诞生的计算机行业。他参与了斯坦福研究所（SRI）和美洲银行研制的第一个计算机银行系统 ERMA，随后作为访问副教授加入了当时正在拼命招人的麻省理工学院，4 年之后拿到终身教职（tenure）。他最显赫的成就就是对话程序 ELIZA。现在对话程序有一个更流行且形象的词儿“聊天机器人”（chatbot 或 chatterbot），但根儿都在 ELIZA。



魏森鲍姆（1923—2008）

ELIZA 本是萧伯纳剧本《卖花女》（*Pygmalion*）中女主的名字，女主是乡下人，颜值底子好但讲一口英文方言，最后被语言学家希金斯调教成了一副贵族腔调。该剧后来被改编为音乐剧《窈窕淑女》（*My Fair Lady*），进而改编为电影。皮格马利翁（Pygmalion）是希腊神话中的塞浦路斯国王，喜独居，擅雕刻。神话的作者和主人公估计都是处女座宅男：男主爱上了自己用象牙雕刻作品中的理想少女，最终少女被痴情感动，变成了真人下凡。这是很多画家和雕塑家钟爱的题材，萧翁那故事也很容易成为某些变态语言学家的精神寄托。



法国画家让–里奥·杰洛姆（Jean-Léon Gérôme）的 *Pygmalion and Galatea* 是同类题材中最出名的，现藏于纽约大都会博物馆

下面是一段 ELIZA 作为心理医生和一个高中女生作为病人的对话。

病人：Men are all alike.（男生都一个德行。）

医生：In what way?（哪个德行？）

病人：They're always bugging us about something.（他们总在烦我们。）

医生：Can you think of a specific example?（你能说个具体的例子吗？）

病人：Well, my boyfriend made me come here.（是我男朋友逼我来的。）

医生：Your boyfriend made you come here?（你男朋友逼你来的？）

病人：He says I'm depressed much of the time.（他说我总是抑郁。）

医生：I'm sorry to hear that you are depressed.（很遗憾听到你抑郁。）

病人：It's true that I am unhappy.（我真的很不爽。）

医生：Do you think coming here will help you not to be unhappy.（你觉得到这来能帮你走出困境（不是不爽）吗？）

病人：I need some help, that much seems certain.（我需要帮助，这是肯定的。）

医生：What would it mean to you if got some help?（你得到帮助，对你意味着什么？）

病人：Perhaps I could learn to get along with my mother.（也许我能学会和我妈相处。）

医生：Tell me more about your family.（说说你的家庭。）

病人：My mother takes care of me.（我妈照顾我。）

医生：Who else in your family takes care of you.（你们家还有别人照顾你吗？）

病人：My father.（我爸。）

医生：Your father?（你爸？）

病人：You are like my father in some ways.（你有点像我爸。）

医生：What resemblance do you see?（哪像？）

病人：You are not very aggressive but I think you don't want me to notice that.（你不咄咄逼人，并且你不想让我注意到这一点。）

医生：What makes you think I am not very aggressive?（你为什么认为我不咄咄逼人？）

病人：You don't argue with me.（你不和我吵。）

医生：Why do you think I don't argue with you?（你为什么认为我不和你吵？）

病人：You are afraid of me.（你怕我。）

医生：Does it please you to believe I am afraid of you?（认为我怕你，会让你高兴吗？）

病人：My father is afraid of everybody.（我爸谁都怕。）

医生：What else comes to mind when you think of your father?（当你想你爸时，脑海还会浮现什么？）

病人：Bullies.（仗势欺人。）

医生：Does that have anything to do with the fact that your boyfriend made you come here?（这和你男朋友逼你来这，有啥关系吗？）

ELIZA 刚刚面世的几年，曾经放在麻省理工学院人工智能实验室对外开放，很多来访的学术界和新闻界的人都要试试和 ELIZA 走一个疗程。很多人都觉得像是刚和一个真的心理医生聊完一样，心情沉重。其实 ELIZA 是一个超级简单的程序，任何一个严肃的计算机科学家都不觉得这个东西有什么智力的难度。它就是在一个按词频排序的词库里找一个词的匹配，如果找到就在脚本库里选择合适的回复。但这样一个简单的东西居然迷惑了很多人，魏森鲍姆的秘书在程序刚面世时就上了瘾，每次在终端上和 ELIZA 聊天时，都要叫旁人离开，这样她可以不用怕流露真情而难为情。这个程序也欺骗了一些人，差点耽误正事。一次魏森鲍姆接待一位 BBN 的副总，副总在终端上玩了一会儿以为真是在和一个人聊天，于是走时打了一句“给我打个电话，号码是 401-1850”，结果终端没回复，把副总气得够呛，其实是他最后一句话没打句号，机器不是傲慢，而是一直在等他结束输入。

20 世纪 50 年代末到 60 年代是斯坦福大学突飞猛进的时代，一堆能人加入，和人工智能相关的除了计算机系的麦卡锡和费根鲍姆外，还有诺贝尔生理奖得主李德伯格（Joshua Lederberger）、化学家翟若适（Carl Djerassi）等。科尔比（Kenneth Colby）也是其中之一，他在医学院教“心理分析”。他的兴趣是把弗洛伊德理论和当代物理学、认知科学和科学哲学挂钩。费根鲍姆介绍科尔比认识了当时在斯坦福大学访问的魏森鲍姆。科尔比感兴趣的问题正好和魏森鲍姆相反：怎么样构造一个能聊天的病人，一方面可以培训心理医生，另一方面理解病人的征兆。他的成果在 1972 年变成了计算机程序 PARRY。下面是 PARRY（病人）和医生的一段对话。

医生：Why are you in the hospital?（你为什么在医院？）

病人：I shouldn't be here.（我不该在这啊。）

医生：Who brought you here?（谁把你带来的？）

病人：Police.（警察。）

医生：What trouble did you have with the police?（你给警察找什么麻烦了吗？）

病人：Cops don't do their job.（警察失职。）

医生：What do they do instead?（他们干啥失职的事了？）

病人：Cops arrest the wrong people.（他们拘错人了。）

医生：They arrest the right ones sometimes.（他们有时也拘对人啊。）

病人：The law should get those Italian crooks.（法律应该收拾那帮意大利小混混。）

医生：Are you Italian?（你是意大利裔吗？）

病人：My background is British but I was born in this country.（我祖上是英国人，但我是在美国生的。）

医生：Do you know any Italian crooks?（你认识意大利混混吗？）

病人：I try to avoid the underground.（我是躲着黑社会的。）

医生：Are you afraid of the underground?（你怕黑社会吗？）

病人：The Mafia are the worst gangsters.（黑手党是最坏的匪徒。）

医生：Are you connected with the Mafia?（你和黑手党有关系吗？）

病人：You know, they know me.（你知道，他们认识我。）

医生：How do they know you?（他们怎么会认识你的？）

病人：They spy on me.（他们跟踪我。）

医生：Why do they?（为什么？）

病人：They want to get false evidence to frame me.（他们在找假证据，给我下套。）

医生：What do they have against you?（你有什么把柄在他们手里吗？）

病人：They are out to get me.（他们要收拾我。）

医生：What for?（为什么？）

病人：I have incriminating evidence against the Mafia.（我有能让黑手党蹲监牢的证据。）

医生：Are you sure of all this?（你肯定吗？）

病人：You don't believe me?（你不相信我吗？）

医生：I am doing my best.（我尽力。）

病人沉默。

心理医生沉默。

病人：Can't you think of a decent question to ask?（你能想出一个靠谱的问题来问吗？）

医生：You seem to hostile to me.（你有点敌意啊。）

病人：Some of your remarks annoy me.（你的问题很讨厌。）

医生：I am sorry.（抱歉。）

病人：Forgot the excuses, you guys are all alike.（别找借口了，你们都一个德行的。）

1972 年的国际计算机通讯年会（ICCC）上，通过互联网的前身 ARPANET，位于波士顿郊区 BBN 公司的 ELIZA 作为医生，和位于斯坦福大学人工智能实验室的 PARRY 作为病人，聊了一回。所有的对话记录现在都保存在位于硅谷的计算机历史博物馆里。

从 ELIZA 和 PARRY 分别的表现来看，现在的小冰等聊天机器人也没

进步很多，但知识库的增大使得现在的聊天机器人更加实用。2014 年，《纽约时报》记者纽曼（Judith Newman）撰文讲述她患自闭症的儿子嘎斯（Gus）在和苹果 Siri 的聊天过程中增强了和现实社会打交道的经验。随着 ELIZA 在媒体上的成功，魏森鲍姆和科尔比渐行渐远。ELIZA 的广泛影响力最终把魏森鲍姆变成技术的异教徒。当大家在争论机器能不能最终超越人类时，他指出这不是“能不能”的问题，而是“该不该”的问题。他意识到了技术的可怕，此后的研究方向转变成计算机技术所带来的社会责任和道德问题。

4. 维诺格拉德和积木世界

维诺格拉德（Terry Winograd）本科是在科罗拉多学院（一家文理学院）读的，主修数学，在去麻省理工学院读研究生之前去伦敦大学学院（UCL）干过一年富布莱特学者，领域是语言学。1967 年他来到麻省理工学院时的老师是明斯基和佩珀特。那时细胞自动机和机器人是人工智能实验室的两个热门方向。



维诺格拉德（1946—　）

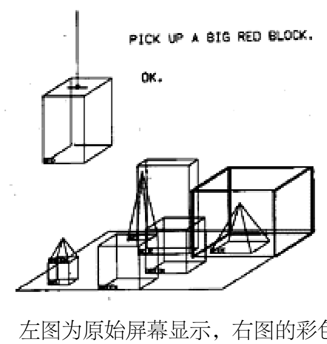
大学中的研究不一定能够成为产品，但必须有实现。实现和产品的距离即使在计算机科学中也是不同的，对于计算理论的人，只要能在理论模型上讲得通就行了，例如图灵机、RAM 和 PRAM 模型等；对于做人工智能的来说，能不能演示（demo）是实现的标准。学术界有所谓“要是发不了文章就滚蛋”（publish or perish）的说法，到人工智能圈里就演变成了“要是演示不了就去死”（demo or die）。这种文化的“始作俑者”是麻省理工学院的人工智能实验室。明斯基喜欢演示，他的个性为人工智能实验室的文化定了型。对于做系统的人来说，只能演示仍然不够，光说不练假把式。明斯基曾说过“一个程序就是需要被 debug 的一张白纸”（a program is just an empty page that needs to be debugged），对于计算机科学的不同子领域，这张白纸的空白程度不一样。对于理论家，这张白纸上的东西简单但可用；对于系统建造者，这张白纸充实但也可用；而对于 AI 研究者，这张白纸苍白，而且上面的东西杂乱且不一定可用。麻省理工学院的计算机实验室（LCS）和 AI 实验室在同一栋楼里，但 LCS 在五层而 AI 在八层，第九层则是计算中心。在斯坦福大学，AI 实验室根本就在校园外，AI 的教授只是在教授会议时才到计算机系来一趟。AI 和主流计算机科学的关系，在学科的早期，是若即若离的。在麻省理工学院，两伙人之间存在严重的敌意；而在斯坦福大学，两伙人之间是互相看不起的疏远。

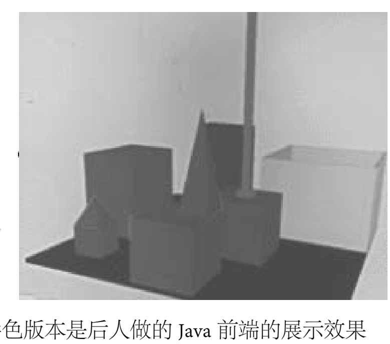
明斯基设计的机器手像是龙虾的爪子，“手眼协调”（hand-eye）和儿童的故事理解是他感兴趣的课题，这都和语言有关系。维诺格拉德准备博士论文题目时，为了理解儿童世界，还借了几本儿童百科全书研读，后来他发现这是一条错误路线，儿童的知识还是太深，需要有更简单的语言世界。这个更简单的世界就是后来被称为“积木世界”的东西。维诺格拉德给他的系统取名 SHRDLU，发音近似“谢德撸”。这名字源于某一种键盘

上字母的排列顺序，英文字母按照使用频率排序是 ETAOINSHRDLU，这种键盘就把 ETAOIN 放在第一排，SHRDLU 放在第二排，这种键盘有时也叫 ETAOIN 键盘，有别于常规的 QWERTY 键盘。

维诺格拉德到麻省理工学院的第一年，在语言学系听了几门句法的课程，后来发现乔姆斯基的语言学系和明斯基的 AI 实验室势不两立。他上了一门乔姆斯基亲授的句法课程，期末论文中他试图说明为什么人工智能的方法可行，老乔居然给了他一个 C，弄得他再也不敢选语言学课程了。

当时真正建立一个机器人手臂，还有一定难度，于是维诺格拉德利用显示器展示了一个虚拟的积木世界。SHRDLU 为这个积木世界建立了模型，例如每个积木块的性质，长宽高、颜色、形状等，以及什么积木块可以搭在什么积木块的上面。人可以通过简单的自然语言，命令一个机器手对这个积木世界进行虚拟操作。例如拿起一个特定的积木块把它摞在另一个积木块上。当机器吃不准人的命令时，可以向人发问。在 20 世纪 70 年代末，彩色显示器还很稀少，维诺格拉德用的是黑白显示器，在每个积木块下标注颜色。后来 SHRDLU 被加上各种前端，目前就有好几个 Java 和 Python 开源前端作为游戏。





左图为原始屏幕显示，右图的彩色版本是后人做的 Java 前端的展示效果

下面就是 SHRDLU 被媒体和教科书引用最多的对话。

人：Pick up a big red block.（拿起大的红方块。）

机：OK.（好。）

人：Grasp the pyramid.（抓住棱锥体。）

机：I DON'T UNDERSTAND WHICH PYRAMID YOU MEAN.（我不懂，哪个棱锥体？）

人：（changing their mind）Find a block which is taller than the one you are holding and put it into the box.（找到那个比你手里的积木高的方块，把它放到盒子里。）

机：BY "IT", I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE ONE I AM HOLDING.（你说"它"是指比我手里这个更高的那个方块吗？）

机：OK.（好了。）

人：What does the box contain?（盒子里有什么？）

机：THE BLUE PYRAMID AND THE BLUE BLOCK.（蓝色棱锥体和蓝色的方块。）

人：What is the pyramid supported by?（棱锥体靠什么支撑的？）

机：THE BOX.（盒子。）

人：How many blocks are not in the box?（盒子外有几个方块。）

机：FOUR OF THEM.（4 个。）

人：Is at least one of them narrower than the one which I told you to pick up?（至少有一个要比我告诉你拿起的那个要窄吗？）

机：YES, THE RED CUBE.（是的，红的立方体。）

人：Is it supported?（它是有依靠的吗？）

机：YES, BY THE TABLE.（是的，靠在桌子上的。）

人：Can the table pick up blocks?（桌子能拿起方块吗？）

机：NO.（不能。）

人：Can a pyramid be supported by a block?（棱锥体能放在一个方块上方吗？）

机：YES.（能。）

人：Can a pyramid support a pyramid?（一个棱锥体能放在一个棱锥体上吗？）

机：I DON'T KNOW.（我不知道。）

人：Stack up two pyramids.（把两个棱锥体摞起来。）

机：I CAN'T.（我不能。）

人：Will you please stack up both of the red blocks and either a green cube or a pyramid?（你能把两个红方块和或者一个绿正方体或者一个棱锥体摞起来吗？）

机：OK.（好。）

人：Which cube is sitting on the table?（哪个正方体还在桌子上？）

机：THE LARGE GREEN ONE WHICH SUPPORTS THE RED PYRAMID.（大的绿色并且正在支撑着红的棱锥体的那个。）

SHRDLU 要远比 ELIZA 复杂，学术意义也更加深刻。SHRDLU 把当时很多 AI 技术整合到一起，除了自然语言理解外，还有规划（Planning[①]）和知识表示[②]。这甚至是最早的计算机图形学的应用。SHRDLU 的潜在应

① 事实上，维诺格拉德为此项目专门设计了编程语言 Micro-Planner。

② 可以争辩这是第一个实用的框架实现，框架要解决的一个主要问题是当前场景到下一场景的状态变化。

用范围也要远广于 ELIZA，我们可以轻易地联想到怎么把 SHRDLU 推广到不同的领域，例如计算可行的旅游路线。

积木世界也有更深的哲学意义。维特根斯坦后期哲学中的语言游戏的理论片段可以在 SHRDLU 的框架下得以更清晰的阐述，尽管维诺格拉德承认他对 SHRDLU 的哲学认识是马后炮。积木世界涉及了语言的好几个方面：语言的输入输出和生成，知识表示和理解，世界和思想。维特根斯坦在《哲学研究》开头所说的奥古斯丁的原始语言。建筑工和他的学徒工关于怎么递送建筑材料的对话场景和积木世界的复杂性没多大差别。维特根斯坦说意义就是语言的使用。积木世界就是语言游戏，是研究语言的一个方法。语言的使用就是心（mind）和物（world）之间的交互。

那年头，用实际的机器人来实现积木世界，成本太高。但我们可以想象，把 SHRDLU 架到一个实际的机器人上是可行的。计算机图形学由此得到有趣的发挥和应用。

SHRDLU 为维诺格拉德赢得了 IJCAI（国际人工智能联合会）1971 年颁发的第一届"计算机与思维"奖。和魏森鲍姆类似，维诺格拉德在功成名就后，也离开了自然语言理解领域，但理由却恰恰相反：维诺格拉德认为人工智能不能取得开始的预期。他在 20 世纪 70 年代初一度与德雷弗斯（Hubert Dreyfus）和塞尔（John Searle）这两位人工智能的批判家混在一起，他们仨每周午餐一次，或在维诺格拉德所在的斯坦福大学，或在两位哲学家的大本营加州大学伯克利分校。两位哲学家非常珍惜这个内行。维诺格拉德干一行不爱一行，他对人工智能的批评，有点像他的计算机系里非人工智能专业的同行，不是挑刺，而是压根看不起。他自己的研究方向逐渐转向人机交互。他的两个学生布林（Sergey Brin）和佩奇（Larry Page）创办了谷歌。

值得一提的是，维诺格拉德曾和人合作过一本书 *Understanding Computers and Cognition: A New Foundation for Design*，这是他研究人机交互的成果。而该书的合作者弗洛雷斯（Fernando Flores）是位传奇人物，他在智利第一位民选总统、社会主义者阿连德任内曾出任财政部长。皮诺切特篡权后，弗洛雷斯蹲了三年大牢，在大赦国际的帮助下，他后来离开智利，带着他的乌托邦理想来到硅谷核心城市、斯坦福大学所在地帕罗阿托，并结识了维诺格拉德。他后来在加州大学伯克利分校得了哲学博士学位。一点也不惊奇，他的指导教师是德雷弗斯兄弟和另一位哲学教授塞尔，论文题目是 Management and Communication in the Office of the Future，据说是受到海德格尔的影响。还好这说的是未来，要不大概只有阿伦特知道海德格尔怎么管理 Office。



弗洛雷斯（1943—　）

5. 统计派又来了

整个 20 世纪 80 年代，自然语言处理的研究乏善可陈。在 1988 年的计算语言学会议上，IBM TJ Watson 研究中心机器翻译小组发表了统计机器翻译的论文，并推出法语/英语的翻译系统 CANDIDE，这标志着统计派

在大数据的支持下又回来了。两年后，同一小组又在《计算语言学》杂志发表论文对他们的工作做了更加理论性的概括。他们的语料是加拿大议会的会议纪要。这两篇划时代的文章虽不长，却有 8 个作者之多，贾里尼克（Frederick Jelinek）是这个小组的核心。贾里尼克的学术训练是信息论，统计是他们这一派人最自然的工具。他的金句是："我每开除一名语言学家，我的语音识别系统的性能就提高一点。"火药味十足。



贾里尼克（1932—2010）

IBM 小组的成员之一柯克（John Cocke）因为 RISC 架构在 1987 年就得了图灵奖。也正因此，他在其他方面的贡献被忽视了，例如，他就是著名的上下文无关文法分析算法 CYK 的"C"。他在图灵奖的致辞中说，计算机性能的提升主要源于三个方面：算法、编译器和体系结构。这三个方面是按重要性大小排序的，但他的名声却主要来自于他认为重要性最小的体系结构。毫无疑问，他对自然语言的兴趣来自他对编程语言的关注。

其实最早提出机器翻译的韦弗（Warren Weaver）的思路就是统计。但乔姆斯基登场后，统计方法基本就没饭吃了。乔姆斯基的理由很简单，语言的可能性是无限的，统计不可能解决问题。乔姆斯基对统计方法的排斥，恰似波普尔对卡尔纳普归纳法的批判。乔姆斯基不喜欢统计派的一个理由是他们太像行为主义了：在翻译的统计方法中，平行语料的左边就是刺激，右边就是反射。

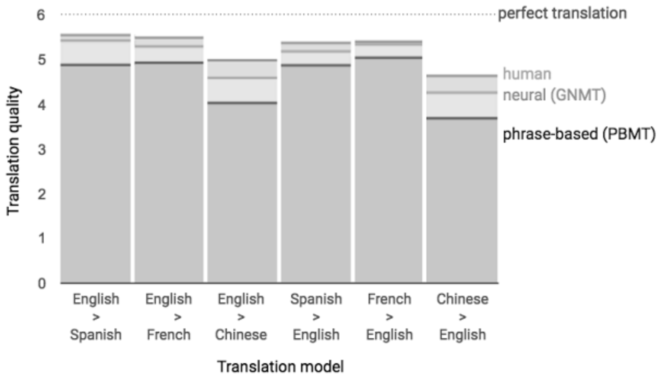
CANDIDE 虽有突破，但相较传统方法，性能优势并不明显。2004 年，已在南加州大学的信息科学研究所（ISI）干了两年博士后的德国人欧赫（Franz Josef Och）加入谷歌。他在亚琛工业大学的博士论文就是用大量平行语料构建语言模型和翻译模型。谷歌海量的数据让欧赫如鱼得水。谷歌翻译器迅速成为行业标杆，在 NIST 随后组织的机器翻译测试中一直名列前茅。2014 年欧赫在谷歌呆了十年后先后加入两家基因测序公司。

统计方法的另一个好处是工程师根本不需要语言学知识，也不需要懂源语言或目标语言，就可从事机器翻译。谷歌翻译团队就没什么科班出身的语言学家。欧赫认为语言学知识对翻译没什么用处，有时还会起反作用。

6. 神经翻译是终极手段吗？

2016 年，谷歌发布神经机器翻译（Google Neural Machine Translation，简称 GNMT）系统，再次大幅提高机器翻译的水平。和谷歌更早期的基于短语的机器翻译（Phrase-Based Machine Translation，简称 PBMT）不同，神经翻译的基本单位是句子。谷歌使用了循环神经网络 RNN 做序列到序列（Sequence to Sequence）的学习，硬件设备是谷歌自己的 TensorFlow 平台。神经翻译相比谷歌早期的基于短语的翻译系统，误差降低了 60%，这是翻译质量巨大的提升。谷歌的这项工作已经开源。

2017 年，Facebook 进一步提高了翻译效率。他们用自己擅长的卷积神经网络 CNN，进行序列到序列的学习。Facebook 号称，英文–德文和英文–法文翻译的基准测试表明，他们的结果在准确度上不输谷歌，而在计算速度上则可以比谷歌的 RNN 有一个数量级的提升。



神经翻译和早期翻译方法的质量对比

RNN 和 CNN 两种神经网络架构，分别被谷歌和 Facebook 支持。性能的此消彼长也被视为两家公司的竞争。真难预料神经网络还有多大的潜力可以挖掘。

乔姆斯基们也许会接着质疑，这种翻译算理解吗？也许翻译根本就不是理解的问题，翻译本身并不需要解释，翻译只是翻译而已，翻译只是数据问题，而不是语义问题。没有乔姆斯基，我们还要在黑暗中摸索，但有了乔姆斯基，是不是又曾经束缚了我们探索其他方法的可能性？

7. 问答系统和 IBM 沃森

语义的问题大致可分两类，一类体现在人和世界的互动，例如维特根斯坦在《哲学研究》里举的例子。建筑工地上，师傅叫徒弟：“给我递块砖。”徒弟真给递块砖过来，那徒弟就是听懂了语义，维诺格拉德的积木世界属于这类。还有一类体现在人和人的交流。魏森鲍姆的 ELIZA 就属这类。现在的问答系统比积木世界和 ELIZA 已经复杂很多。

大体上说，问答系统有三个必备的组成部分，第一部分是问题理解，第二部分是知识查询，第三部分是答案生成。这三个部分相辅相成，第一部分和第三部分是自然语言处理的工作，它们通过知识图谱被有机地整合在一起，在定理证明火的时候，被当成狗皮膏万能灵药。问答被转换成定理证明过程。正是在问答系统的研究中发现了定理证明方法在知识表示上的局限。现在的问答系统依靠常识和知识，同时也依靠浅层的推理。知识图谱是核心。

一个问题总是会涉及 who，when，where，how，why。像我们小学作文课教的：一个事件要有人物、时间、地点、关系、因果等。这些要素都可以套到知识图谱中类和实体的属性和关系上。当你问 2016 年之后的搜索引擎："梁启超的儿媳妇是谁？"答案中至少会有"林徽因"和"林洙"。道理很简单，因为系统底层的知识图谱知道梁思成是梁启超的儿子，而梁思成结了两次婚，第一次林徽因，第二次林洙。

当知识图谱足够大的时候，它回答问题的能力会惊人。2011 年 IBM 的沃森（得名于 IBM 创始人 Watson）在美国电视智力竞赛节目 *Jeopardy!*（《危险边缘》）中击败人类选手，并获得百万美元大奖。沃森的知识图谱包括 WordNet、Dbpedia 和 Yago。Dbpedia 和 Yago 都是以维基百科为基础数据源的常识图谱，在此之上，沃森还搜集了 IMDB 电影数据库等各种专业数据源。

在 *Jeopardy!* 节目中出现过的问题，95%都能在维基百科中找到答案。沃森参赛的版本的知识库只有 4TB，其中包含了所有维基百科的正文，真的不大。除了半结构化的知识图谱，沃森还使用了开源搜索引擎。把搜索的结果文档的标题与维基百科词条进行匹配，如果在维基百科中能找到，就把搜索结果列入候选答案。再把候选答案反馈给搜索引擎，进一步对返

回结果做证据支持的处理，然后给出答案。

硬件系统是一个有 90 台 IBM Power 750 的集群，每台配一个 IBM Power 78 核处理器，每核 4 线程，所有一共 720 核，2880 线程；内存 16TB，所有的知识图谱都放在内存里了。按照 Linpack 基准程序，这台计算机的算力相当于当年排名第 500 的超级计算机的一半，成本只有 300 万美元。同沃森带来的巨大广告效应相比，这真不算什么。按照摩尔定律，在本书出版时（2017 年），同等算力大概只需要不到 20 万美元。

IBM 吸取了深蓝的教训，沃森在 *Jeopardy!* 节目上取得的宣传成功后，很快变成了 IBM 人工智能事业的品牌，IBM 很快推出了沃森金融、沃森医疗、沃森教育等。现在 IBM 整个公司都围绕沃森转型了，也许 IBM 觉得“人工智能”这个词儿太俗了，他们非要标新立异地自诩为“认知计算”。

8. 回顾和展望

在谷歌和 Facebook 在机器翻译取得长足进展之前，语音输入率先获得突破，2011 年微软邓力团队在语音识别上的成功是深度学习的标志性事件之一。苹果、谷歌和微软都推出实用产品。中国企业科大讯飞的语音输入的应用技术也渐趋成熟。

1976 年，卡内基梅隆大学的瑞迪（Raj Reddy）教授曾大胆预言十年内，可以用两万美元造一台语音识别系统。但实际上花了差不多 35 年，这个延迟和司马贺在计算机下棋预言的延迟差不多。瑞迪因为他在语音技术的贡献而得了图灵奖。但平心而论，瑞迪的学术思路和后来取得成功的神经网络技术完全是对立的。瑞迪早期在卡内基梅隆大学领导的语音识别系统 Hearsay 是符号派的成果，由此衍生出来的“黑板系统”（Blackboard）曾是

人工智能的热门架构之一。瑞迪的学生莱瑟（Victor Lesser）是“黑板系统”的原创者，后被阿比卜挖到麻省大学，一度是人工智能领域的热门人物。随着专家系统的衰败，“黑板系统”也日渐凋零，莱瑟后来把“黑板系统”改名叫“多 Agent 系统”，这个新名词现在是一票残余的符号派的保护伞。语音识别的成功确实和符号派没啥关系。历史真是有点讽刺。

语音的同声翻译成为实用，也是指日可待的事情。就像一个哲学问题找到了科学的角度（formulation），就不再是哲学问题一样，一个人工智能问题一旦解决，就不再是人工智能问题。大概很快人们就会认为语音问题不再是人工智能的核心问题。如果说语音翻译不涉及自然语言理解和语义，可能也不会有什么异议。

随着各种神经网络技术的引用，机器翻译的水平提升得很快。预言家库兹韦尔说，2029 年机器翻译就可达到人的水平。按照当下神经翻译的进展速度，也许这个目标是现实的。乐观主义和悲观主义都有前例可循。但是即使自然语言翻译问题得到了解决，机器可以算是“理解”吗？也许我们根本不需要讨论“中文屋”问题，当机器翻译问题得到解决时，“中文屋”问题自动就成了伪问题呢。

2011 年 5 月，麻省理工学院为配合 150 周年校庆，召开了“大脑，心，机器”的研讨会（Brain, Mind and Machine Symposium），把本校的几位大佬都请来，乔姆斯基、明斯基、温斯顿等悉数出席，并由语言学家平克（Steve Pinker）主持。乔姆斯基批评当下流行的神经网络和统计方法，乔姆斯基认为神经网络是黑盒子，并没有给我们提供解释，故而没有提供知识。麻省理工学院主办的《技术评论》杂志为这个研讨会发了专文，标题是故意挑事儿的“不会思维的机器”（Unthinking Machines）。时任谷歌研发总监的诺维格（Peter Norvig）很快回应乔姆斯基，他批评语言学的规则在自然语言

处理上，根本就没用。有人开始用“两种文化”[1]来总结乔姆斯基和诺维格的隔空掐架。

乔姆斯基对人工智能的批评的核心在于“可解释性”。AlphaGo 不能解释自己下棋的路数，算不算会下棋呢？也可以反过来说，只有解释了，人类才能从中得到洞见，学习知识。但解释是不是也有层次，只有学会牛顿力学，才能学会相对论和量子力学？就如维特根斯坦所说的梯子的比喻，爬上房顶，梯子才能扔掉，梯子就是解释。

其实，即使人类在不理解力学的时候，就会造弹弓了。对那时的人类，弹弓的工作原理就是黑匣子。乔姆斯基和诺维格分别所代表的两种人关心的是两种不同的问题。一种人力图打造实用的工具，没有解释也能凑合，他们是不求甚解的工程师；另一种人寻求终极的知识，他们是科学家。只不过，在计算机科学这个特定的学科中，科学家和工程师的角色变换太快，这门学科的开拓者，很多都是身兼二职，例如图灵和冯诺伊曼。

参考文献指南

乔姆斯基的《句法结构》是本小册子，不需要什么背景就能读。Brown (1988，1990)是统计派的奠基作品，正文只有 6 页，虽是学术论文，却非常可读。

Winograd (1972) 描述了 SHRDLU 的工作原理。维诺格拉德花了 5 年时间准备撰写一部三卷本的全方位的自然语言理解的教科书，但该书出了第一卷（Winograd (1982)）之后，他就放弃了——写书太耗神。这样前功半

① “两种文化”（*The Two Cultures*）是英国科学家兼小说家 Charles Percy Snow 一本书的名字，特指科学和人文两种文化的对立。

弃的例子还有很多，丘奇的《数理逻辑》也是只有第一卷。罗素在完成《数学原理》第三卷后对学术的疲惫，就像高中生高考之后对书本的痛恨。

巴贝奇研究所的口述历史是有趣的信息来源，其中，Nilsson (1989)、Reddy (1991)和 Winograd (1991)同本章最为相关。大部分人工智能相关的口述历史的采访都在 1990 年统计派重新主导之前。Hutchins (2005)是一段可读的机器翻译"简"简史，但没有提及统计方法。

IBM Journal of Research and Development 是一本高水平的学术刊物，主要登载 IBM 研究人员的成果，但也时有例外，曾经能自由浏览，但 2010 年起每期只有头几篇文章公开，其余则需要订阅，该杂志网页首页给出了曾经登载过的那些图灵奖和准图灵奖的文章的链接，例如 RISC、SQL、算法信息论、量子计算等，大概是 IBM 研究部门做给他们老板看的。当年贝尔实验室还在的时候，也有本杂志 *Bell System Technical Journal*，上面也曾发表过重量级的文章，例如香农的 A Mathematical Theory of Communication，以及 UNIX 的原始文章等。但随着美国电信业反垄断法的实施，贝尔实验室四分五裂，杂志也屡改其名，几经转手，不知道还有没有读者了。话说回来，对沃森有进一步兴趣的，可参阅 2012 年的 *IBM Journal of Research and Development* 5/6 月合刊，那是沃森专辑，文章还不算过气。

当人工智能先驱明斯基去世时，《连线》（*Wired*）杂志用 Automated Insights 公司的写作机器人 Wordsmith 为明斯基写了一段讣告，虽不如《纽约时报》讣告栏的那么文采飞扬，但也中规中矩。好在这只是给一个人写的讣告，而不是给全人类的。

Marvin Lee Minsky, 88, passed away January 24, 2016 in Boston, Massachusetts of cerebral hemorrhaging.

Born August 9, 1927 in New York City, New York, to parents Fannie Reiser

and Henry Minsky, Marvin Minsky was known for his pioneer contribution to the field of artificial intelligence (AI). After graduating from Phillips Academy, Minsky attended Harvard University, graduating with a BA in Mathematics in 1950. He continued his education at Princeton University, ultimately graduating with a a PhD in Mathematics in 1954.

Some of Minsky's greatest accomplishments include founding the MIT Computer Science and Artificial Intelligence Laboratory in 1959 and authoring many groundbreaking books in the field of artificial intelligence, including Perceptrons. He won many notable awards in his field of study, including the Turing Award in 1969.

Minsky is survived by his wife Gloria Minsky; three children, Margaret Minsky, Julie Minsky, and Henry Minsky.

其实有了靠谱的知识图谱，这类的八股文章是信手拈来，水到渠成，都是套路。这段机器的话再用搜狗的翻译器（2017 年 8 月版）译成汉语：

88 岁的马文·lee 于 2016 年 1 月 24 日在马萨诸塞州波士顿去世。

1927 年 8 月 9 日生于纽约市纽约市的父母范妮·赖泽和亨利·明斯基，马文·明斯基以他对人工智能领域的先驱贡献而著称。在从菲利普斯学院毕业后，明斯基就读于哈佛大学，1950 年毕业于数学学士学位。他继续在普林斯顿大学教育，最后于 1954 年获得数学博士学位。

明斯基的一些最伟大的成就包括于 1959 年成立麻省理工学院计算机科学和人工智能实验室，并在人工智能领域，包括认知机领域创作了许多开创性书籍。他在他的研究领域赢得了许多著名的奖项，包括 1969 年的图灵奖。

明斯基被他的妻子凯莱明斯基（gloria 明斯基）幸存；三个孩子，玛格丽特·明斯基、朱莉·明斯基和亨利·明斯基。

第 8 章
向自然学习：从遗传算法到强化学习

Natural selection is a mechanism for generating an exceedingly high degree of improbability.

自然选择就是能生成极不可能之事的机制。

——Ronald Fisher（费舍）

从生物学里找计算的模型，一直是人工智能的研究方向之一，学术上大致有两条传承的脉络：一条是麦卡洛克和皮茨的神经网络，演化到今天成了深度学习；另一条是冯诺伊曼的细胞自动机，历经遗传算法、遗传编程，其中一条支线最后演变成了今天的强化学习。

1. 霍兰德和遗传算法

霍兰德（John Holland）本科在麻省理工学院学物理，毕业后在 IBM 工作了一年半，老板是达特茅斯会议的策划者之一罗切斯特（Nathaniel Rochester）。他后来到密执安大学读研究生，一直给 IBM 干活，所以 4 年学费也都是 IBM 付的。霍兰德在麻省理工学院读书时曾见过维纳，但维纳对他没什么影响，倒是他上过的一门当时称为“数值分析”的课，为他后来转入计算机领域埋下了种子，那时的“数值分析”就是现在的“算法分析”，和现在“数值分析”不尽相同。霍兰德研究生的专业一开始是数

学，那时密执安的数学系教授中有好几个布尔巴基学派的成员以及他们的徒子徒孙，他们和逻辑与代数的关系密切，这也是霍兰德的兴趣所在。

霍兰德在 IBM 做夏季实习生时认识了麦卡锡，是麦卡锡教会他下围棋。后来在 DARPA 位高权重的利克莱德（Joseph Licklider）[①]教会他神经网络。他们还多次到加拿大麦吉尔大学（McGill）拜访在那里任教的赫布（Donald Hebb），现在深度学习的原创思想很多都可以溯源到他。



霍兰德（1929—2015）

霍兰德准备动手写关于代数和逻辑的博士论文时，遇见了在哲学系任教的伯克斯（Authur Burks）。伯克斯是密执安大学的哲学博士，1941 年 25 岁时博士毕业去了宾夕法尼亚大学的摩尔学院，加入美国最早的计算机之一 ENIAC 的研制。冯诺伊曼当时想把整个 ENIAC 团队招安到普林斯顿高等研究院，ENIAC 团队的头儿工程师毛彻里（John Mauchly）被老冯所不喜，老冯看中的是 ENIAC 的工程骨干埃克特（John Presper Eckert），但埃克特不愿背叛毛彻里。于是冯诺伊曼挖到了 ENIAC 项目早期真正的灵魂人物数学家、逻辑学家古德斯丁（Herman Goldstine），伯克斯随着古

① 利克莱德在 DARPA 时资助了人机交互和人机协同，以及 ARPANET（后演变为互联网）。他的名字在人工智能各位大佬的晚年采访中都被多次提到，认为他是领袖式人物，但在人工智能的各种历史中却鲜被提及。这是个有趣的现象。

德斯丁加入到普林斯顿高等研究院团队，先是做老冯的助手，后来参与了美国最早几台计算机的研发。


伯克斯（1915—2008）

战后，伯克斯又回到母校密执安大学教哲学。20 世纪 50 年代中期，王浩曾经到密执安大学访问伯克斯，两人还合作了一篇细胞自动机的文章发表在 1957 年的 *JACM* 上。霍兰德碰见伯克斯的时候，他刚刚创立了通讯科学计划（Communication Sciences Program），这是个哲学、数学、心理学和语言学的跨学科计划，后来演变成密执安大学的计算机和通讯科学系，这个名字直到 20 世纪 90 年代初才改成更常规的“计算机科学系”。这个跨学科计划恰是霍兰德的兴趣所在，于是他变成了伯克斯的学生。他所在的小组叫“计算机逻辑组”（Logic of Computers Group）。霍兰德由此成为有史以来计算机科学的第一个博士。和霍兰德在密执安一起在这个跨学科计划呆过的还有华裔语言学家王士元（William Wang），霍、王二人后来成为好友。不太为人知的是，王士元后来也参与过 IBM 组织的乔治敦机器翻译的工作。

霍兰德比麦卡锡和明斯基小几岁，但基本还算是同一代人。麦卡锡离开麻省理工学院去了斯坦福大学，霍兰德在晚年接受采访时如此评论麦卡锡和明斯基：美国西部的人工智能由麦卡锡代表，他们干净（neat），一切讲究逻辑；东部的领袖自然是明斯基，他们邋遢（scruffy），做事比较随意（adhoc）。但他们的共性是都对机器学习不太感兴趣。尤其是明斯基对罗森布拉特《感知机》的批判导致了神经网络的衰落，从某种意义上也是人工智能第一次低潮的起因，所谓搬起石头砸了自己的脚。霍兰德没有提及人工智能的另一重镇——卡内基梅隆大学，那里的旗帜不那么鲜明。

霍兰德的博士论文题目是 Cycles in Logical Nets。伯克斯也写过本小册子《逻辑网络理论》（*Theory of Logical Nets*）。所谓“逻辑网络”，当时是个模糊的概念。麦卡洛克和皮茨的神经网络模型也称为逻辑网络，因为皮茨本人是逻辑学票友。冯诺伊曼的细胞自动机也是逻辑网络。伯克斯是老冯细胞自动机遗著的编者，霍兰德受老师伯克斯的影响也是自然的。他们学生的博士论文多少都和细胞自动机有关。20 世纪 50 年代是逻辑学逐渐离开哲学，向其他学科渗透的时代，逻辑是一股风气，什么人都喜欢和逻辑沾点边，就像当下的人工智能或深度学习。

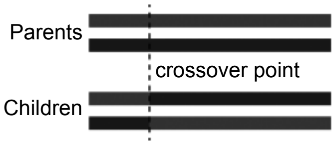
有意思的是，在麦卡锡执笔的达特茅斯会议的计划书里，有一节“神经网络”。其中，霍兰德的名字和麦卡洛克、皮茨、明斯基和罗切斯特等人的名字并列。晚年他回忆说当时确实收到了达特茅斯会议的邀请，但那个夏季他要在密执安教课，就没去，读研究生时找到份夏季工作不容易。估计当时谁也没觉得那个会后来变得如此重要。霍兰德为未能参会颇为遗憾，认为是他个人的重大损失。可不是嘛，那个会的参会者都可自居 AI 的创始人。

霍兰德认为达特茅斯会后 AI 基本就是符号派一统天下了。学习，或者用霍兰德的话说“可适应”（adaptation）作为人工智能的一个重要分支，要到好多年后才翻过盘来。霍兰德说他自己的思想被学界逐渐接受，是在他的学生都出了名之后。美国的师生关系和中国确有不同，美国是学生毕业后，自立门户，大部分还是接着原来的东西继续做，也可以跨越式发展；但在中国，大部分是等着接老师的班儿，老师是院士，就扶持学生当院士，老师是校长，学生接着做校长，一旦一个“重点实验室”建立，小佬坐等大佬死后接班升大佬。

霍兰德在回顾自己的研究生涯时说，如果一个人在早期过深地进入一个领域，可能会不利于吸收新的思想。对于霍兰德来说，进化论和遗传学是新思想，幸运的是他的老师伯克斯也是跨界人才，鼓励交叉学科的研究。对霍兰德影响最大的一本书是英国统计学家费舍（Ronald Fisher）的《自然选择的遗传理论》（*The Genetical Theory of Natural Selection*）。无神论者道金斯（Richard Dawkins）称费舍是达尔文之后最伟大的生物学家。费舍把孟德尔的遗传理论和达尔文的自然选择结合起来。霍兰德由此得到启发：进化和遗传是族群学习的过程，机器学习可以此为模型。

早在 1958 年，IBM 的弗雷德伯格（R. M. Friedberg）就研究过“进化程序”，还在一台 IBM-704 机器上做过模拟。但他的同事马上就指出他的“进化程序”其实比随机搜索还要慢。无论如何，弗雷德伯格的工作给了霍兰德启发，可算得是遗传算法的先兆。

染色体（chromesome）是遗传的基本单位。以人为例，人有两性，男性第 23 对染色体呈 X-Y，而女性只有 X。两性交配导致人类染色体的交叉（crossover）。在进化过程中，部分基因还会变异（mutation）。环境会保留某类基因的族群，而淘汰掉其他的。



染色体交叉，子代分别从父母取得一部分染色体

遗传算法就是模拟种群（population）的进化过程。其结构可以用下列伪代码大致表示。

(1) 随机生成初始群体。

(2) 主循环（停机的标准可以是迭代次数，或者适应度达到某个要求）。

a) 执行策略，计算当前群体中所有个体的适应度；

b) 从当前群体中，选择精英作为下一代的父母；

c) 将选出的精英父母配对；

d) 以极小概率将子代变异；

e) 将子代个体添加到新群体中。

从程序中，我们马上可以理解进化中"优胜劣汰"的算法含义。伴随 20 世纪 80 年代后期神经网络的复兴，遗传算法也作为一种生物学启发（biology-inspired）的算法，得到更多的认可，同时也有更多的实际应用。1985 年第一次遗传算法国际会议召开，这个学科算是有了自己的共同体。1997 年 IEEE 开办了《进化计算杂志》(*IEEE Transactionson Evolutionary Computation*)，遗传算法也算是进入主流了吧。

2. 遗传编程

在遗传算法中，种群是数据，更进一步的想法是：如果种群变成程序的话，进化是不是仍然可行呢？霍兰德的学生寇扎（John Koza）在 1987

年给出了一个思路，并把它命名为“遗传编程”（Genetic Programming）。


寇扎（1943—　）

物理学家多依奇（David Deutsch）用生物进化来类比知识的进化，他是哲学家波普尔（Karl Popper）的粉丝，并常常套用波普尔的科学哲学术语。他说猜想就像变异，批评和实验就像选择，而交叉学科就是配对了。从这个意义上说，知识的增长更像是遗传编程。

遗传编程的结构和遗传算法差不多，一组程序就一个特定的问题给出解答，按照执行结果的好坏给所有程序排序。程序本身也是数据，自然也可以修改。在遗传编程里，变异就是对程序做微小调整。交叉和配对就是将两个表现优异的程序互相嫁接。寇扎后来还引入了“基因重复”（duplication）和“基因删除”（deletion）等生物学概念，以提升遗传编程的效率。

遗传算法本身就需要大量的数据，遗传编程需要的数据量自然更大，这对计算能力提出了新的需求。并行计算机公司 Thinking Machines 在 20 世纪 90 年代初曾经尝试用超级计算实现大规模的遗传编程，公司创始人希利斯（Danny Hillis，明斯基的学生）在 1994 年的 TED 会议上演讲的题目是 Backtothe Future，他颇为自得地谈起用遗传编程自动学会排序算法。但没过多久，Thinking Machines 就倒闭了。1999 年时，寇扎自己搭建了一个 1000 个节点的集群，每个节点是 Pentium-II（奔腾-2），那时搭建集群的

软硬件技术统称 Beowulf，是当下 Hadoop 和 Spark 的先驱。

遗传算法的稳定性一直就是研究课题，遗传编程的数学性质自然更加复杂。寇扎等人给国际机器学习大会的投稿多次被拒，理由是遗传编程的性能常常还不如一些简单的搜索算法，在大规模的实际问题上无法实用。现在看，这一点也不惊人，其实如果没有算力的大幅提升，眼下红得发紫的各种深度学习都无法实用。寇扎联合遗传算法的人马自己开办了“遗传与进化计算会议”（Genetic and Evolutionary Computing Conference）。

1995 年，寇扎利用遗传编程做布尔电路优化，取得成功，算是遗传编程可实用的里程碑。寇扎 1999 年创业，公司名就叫“遗传编程”。公司是研究型公司，主要为政府和企业提供关于遗传编程的咨询服务。

寇扎说遗传编程是“发明机器”（inventing machines），有了遗传编程就不需要其他人工智能了，他的理由是人工智能的目的是生产有智能的程序，这不正是遗传编程干的吗？听起来有道理，但遗传编程的理论基础一直欠缺。

遗传算法和遗传编程这一脉，在神经网络低谷时，虽然也受到波及，但并没有像神经网络那样备受打击。而在神经网络咸鱼翻身后，也没有爬得那么高。

3. 强化学习

巴托（Andy Barto）在霍兰德手下得了博士后[①]就被麦克·阿比卜（Michael Arbib）招到麻省大学计算机系，此时的麻省大学正在阿比卜领导

① 事实上，是霍兰德的另一位学生齐格勒（Zeigler）指导了巴托的博士论文，但霍兰德也是巴托的导师之一。他们的博士论文都和细胞自动机相关。巴托自认受霍兰德影响巨大。

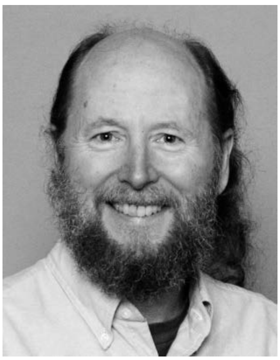
下成为人工智能的重镇，计算机系开始分为理论、系统（包括软件和硬件）和控制论三个方向，而控制论后来成为人工智能。这种分法后来也是所有计算机系的标配。阿比卜一开始坚持“控制论”而拒绝用“人工智能”，有两方面原因，其一，他是维纳最后的学生，并且他终身的学术兴趣是为大脑建模（brain modeling）；其二，“人工智能”这个词儿的流行是在 20 世纪 70 年代中期，按照阿比卜的一家之言：人工智能是控制论的替代品，至少从时间轴上看，这不算错。巴托到麻省大学之后先是做博士后，很快转成助理教授。



巴托（1948—　）

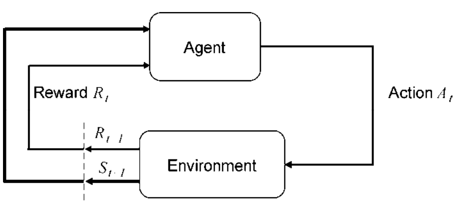
老鼠的孩子会打洞，和他的老师一样，巴托的博士论文研究的也是细胞自动机。巴托到麻省大学任教时，正赶上神经网络的低潮，于是他学老师，把自己的实验室命名为“可适应系统”（Adaptive Systems），听起来要和神经网络保持一定距离。20 世纪 80 年代末的某一学期，麻省大学的人工智能课程由计算机系的所有教授联合开，每人负责一个章节，主讲机器学习的是《人工智能手册》的编辑之一寇恩（Paul Cohen），巴托只是讲讲神经网络。当时很奇怪，他给的参考书是一本名为 *Brain* 的脑科学教科书，课堂里讲的都是生物学，和当时已经开始复兴的神经网络没啥直接关系，学生都颇为不解。

巴托在麻省大学的第一个博士生就是萨顿（Richard Sutton），萨顿本科在斯坦福大学学的心理学，研究动物怎么适应环境一直是他的兴趣。和老师霍兰德不同，巴托和萨顿关心更原始但也更抽象的可适应性。一个刚出生的孩子，怎么学会对环境的适应。在监督式学习中，目标是清楚的。但婴儿不知道目标是什么，不知道自己要什么。通过与外部世界的不断交互，婴儿受到奖励或惩罚，由此强化对外部世界的认知。



萨顿

强化学习的理论基础之一是马尔科夫决策过程。强化学习的主体是 Agent，Agent 和环境互动。在一个时间点 t，环境的表示是当前的状态 S_t，Agent 对环境实施动作 A_t，环境回馈给 Agent 奖赏 R_{t+1}，并导致环境进入一个新状态 S_{t+1}。Agent 的策略用 π 来表示，时间点 t 的策略 $\pi_t(a|s)$就是在状态 $s = S_t$ 时，Agent 采用动作 $a = A_t$ 的概率。强化学习就是 Agent 根据经验改变策略以期达到长期最大奖赏的过程。



强化学习

强化学习的另一个理论基础是动态规划。贝尔曼（Bellman）在 20 世纪 50 年代就发明了动态规划。萨顿和巴托也承认在强化学习早期，受到动态规划的启发。巴托一度在他的强化学习讨论班上让研究生分工研读贝尔曼的经典著作《动态规划》(Bellman 1957)。班上数学好的学生不知所云，算法课里不都有一章讲动态规划嘛，如果强化学习就是动态规划，那还有啥意思？近 30 年后，当强化学习被用来解决围棋这样复杂的问题之后，当年班上的学生们才体会到巴托的初衷。但三十年太久只争朝夕，这是一个人学术生涯的全部。巴托几年前就已经退休了，学生们也到了人生的强弩之末。愚公移山，现在是当时学生们的孩子们的天下，他们赶上好时候了。

在计算能力的约束下，强化学习的环境不宜太复杂。萌芽期的强化学习的例子都是游戏，如贝尔曼的“老虎机”和塞缪尔（Samuel）的跳棋。游戏的环境相对容易定义，在棋类比赛中，环境就是对手和规则。强化学习被用来下围棋不是偶然的。

如果整个世界是完全随机的，那么强化学习就要失效。学还是不学对结果没有什么影响。巴托和萨顿有时也把强化学习称为“享乐主义”（hedonistic），也即学习系统想最大化环境对自己的某种反馈。“享乐主义”这个说法来自于另一位先行者克劳福（Harry Klopf）的一本书名《享乐主义的神经元》(*Hedonistic Neuron*)。“享乐主义”和道金斯的“自私的基因”异曲同工，目的是为类生物（biology-inspired）系统建立基本公理。

强化学习中有所谓“抬头看路”（探索，exploration）和“低头拉车”（苦干，exploitation[①]）之分。探索就是看看有没有别的选择，苦干就是专注于当前的选择。在强化学习中，用希腊字母 ε 表示学习率(learning rate)，值越小，能用于探索的时间就越少，绝大部分时间是在苦干。这就像我们

① exploitation 是个不太好翻译的词，很多人译为“利用”，其实有“被剥削”或“苦干”的意思。

的人生，大部分时间在被压榨，极少时间可以“诗和远方”。如果我们再套用那个戴森的“大鸟”和“青蛙”的比喻：“大鸟”是那些高瞻远瞩的科学家，例如希尔伯特、爱因斯坦、杨振宁等，而“青蛙”是那些埋头苦干解决问题的科学家，例如冯诺伊曼、费曼等。估计“大鸟”们探索的时间比较多，也就是学习率较高。

遗传算法和强化学习有一个共同点：效果要等到多步以后才能看到，这是和监督式学习的主要不同。这就需要尽可能多地访问所有的状态，这样效率就会受到影响。蒙特卡洛模拟是一种减少状态空间搜索的有效办法。最近也有利用深度学习来压缩需要表示的状态空间数目。这还有点意思，本来强化学习初衷是探索生物体学习的模型，现在神经网络又成了强化学习的工具。当状态空间很大时，强化学习可以和蒙特卡洛方法或深度神经网络结合，就使用了蒙特卡洛方法。

强化学习作为机器学习的一个分支，一直没得到重视。谷歌的 AlphaGo 赢了李世石之后，强化学习作为 AlphaGo 的核心算法，一夜之间成为显学。这当然要归功于萨顿和巴托多年的坚持。萨顿在麻省大学博士毕业后去了不远处的 GTE 实验室，GTE 是当年除了贝尔系统之外最大的电话公司。贝尔有个实验室，GTE 当然也得有。萨顿呆在 GTE 实验室的主要原因是方便和老师巴托合作。

巴托的“可适应系统”实验室，在神经网络不景气时，曾经收留过一批无家可归的学术浪人，其中就有吴恩达的老师乔丹。事实上，吴恩达的成名作就是用强化学习来控制无人直升机。为了和巴托合作写他们那本强化学习的经典教科书，萨顿一度回到母校担任“研究科学家”（一种没有终身教职的研究性职位），后来他去了加拿大阿尔伯塔大学（Alberta）计算机系，迅速把那里建成强化学习的大本营。谷歌收购的 DeepMind 团队中最核心的几个人都是萨顿的学生，席尔瓦（David Silver）就是萨顿的

大弟子，而自称“AlphaGo 之手”的黄士杰也曾在萨顿手下做过两年博士后。

2017 年 7 月 7 日，DeepMind 宣布将在萨顿的新巢、加拿大阿尔伯塔大学开办联合实验室，这是 DeepMind 第一次在英国以外设立研究机构。经过多年耕耘，萨顿已经把阿尔伯塔大学建成了强化学习的基地，和计算机系里崇尚游戏的几个教授天作之合，使强化学习在围棋、德州扑克、电玩等领域势不可挡。萨顿的阿尔伯塔之于强化学习，就像辛顿的多伦多之于深度学习，LeCun 的纽约大学之于卷积神经网络。可惜巴托已经退休，强化学习在其发源地美国麻省大学已经无人继承。

萨顿 1979 年到麻省大学跟随巴托和阿比卜，由此开创强化学习。他一直认为强化学习是理解智能的关键。维纳的控制论自问世从没进入过主流，现在更已无人问津。在整个人工智能的各个分支里，大概只有强化学习还留有点儿控制论的影子。

一旦一个算法被天才发明，并成功地在一个领域里得到应用，自然会有二流人才前赴后继把这个算法在其他领域发扬光大。20 世纪 80 年代的神经网络如此，当下的强化学习也如此。谷歌 2017 年用强化学习来寻求 NP-hard 问题的近似解（Mirhoseini et al 2017）。还有人把强化学习和符号方法结合做因果推理（Garnelo 2016）。当然 arxiv 上面的文章从发表到证实，还需要段时间。早年有人质疑遗传算法算不算机器学习，他们认为遗传算法是一种近似优化算法，不能算机器学习。但从某种意义上，任何机器学习算法都是一种优化算法。现在强化学习都被用来求解优化问题了。

如果从写作的角度看，强化学习更像是第一人称叙述，Agent 就是“我”，外部世界（包括他人）都是“环境”。监督式学习更像是第三人称叙述，作者在用一只上帝的眼睛洞察世界，对错分明。第一人称的学习要

比第三人称的学习更本质。罗素（Stuart Russell）和诺维格（Peter Norvig）在他们那本权威且无所不包的人工智能大部头教科书《人工智能：一种现代方法》里说“可以认为强化学习包含了全部人工智能”（Reinforcement learning might be considered to encompass all of AI）。这不无道理。

4. 计算向自然学习还是自然向计算学习

以色列海法大学的进化生物学家 Livnat 和伯克利的理论计算机科学家 Papadimitriou 在 2016 年 11 月的《美国计算机学会通讯》（*CACM*）上发表了一篇封面文章“性作为算法”（Sex as an Algorithm），引起轰动。喜欢的人认为这为进化论找到了新视角，而不喜欢的人则批评杂志的编者和作者是为了博眼球。这篇文章质疑了性在进化中的作用。

哈佛大学的理论计算机科学家、图灵奖获得者瓦连特（Leslie Valiant）曾经从计算的角度研究过机器学习和进化，他把进化当作学习的特例。Livnat 和 Papadimitriou 认为有性繁殖不太容易达到最优点，而无性繁殖才更像是优化算法，他们把遗传算法比作有性繁殖，模拟退火算法比作无性繁殖。



瓦连特（1949—　）

关于性的作用，达尔文没有解释，当代的进化生物学家也没有给出满意的答案。这篇文章的作者认为，尽管性并不是从群体中挑选最优选手的算法，但却可以提高种群的平均水平。性所优化的不是个体，而是混合程度（mixability）。这不仅解释了遗传算法的表现，也引发了人们从计算的角度思考生物学的问题。单从这个意义上，这个讨论就是有意义的。

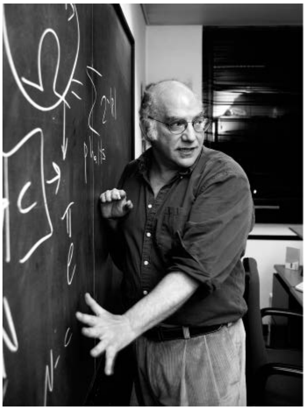
如果说遗传算法是微观地向生物内部机制学习的话，强化学习则是更为宏观地向自然学习。瓦连特的方法企图把微观和宏观整合起来，为学习提供一个更为基础的数学框架。

5. 计算理论与生物学

无论是遗传算法、深度学习还是强化学习，都缺乏计算理论的基础。生物学激发的学科都是模拟自然，它们都不需要解释，不需要了解内部原理，而只要能查看输出结果，就够了。数学大概是所有学科中离生物学最远的学科。

数学家蔡汀（Gregory Chaitin）是 IBM 顶峰期培养的一堆理论家之一。他就读于纽约出名的布朗克斯科学高中，那里曾经毕业过 8 个诺贝尔奖得主。他在纽约城市学院读数学本科时，独立发明了柯尔莫哥洛夫复杂性，最后演化成有他自己特色的“算法信息论”（Algorithmic Information Theory，简称 AIT）。他从 IBM 退休后，回到老家阿根廷，在布宜诺斯艾利斯大学教书，这时他的兴趣转向哲学和生物学。他企图用算法信息论解释进化论，他的成果被他写成一本科普小册子《证明达尔文》[①]。

① *Proving Darwin*，由人民邮电出版社图灵公司引进出版。



蔡汀（1947— ）

蔡汀用数学定义了进化论的几个核心概念。他的数学定义更接近遗传编程而不是遗传算法。首先，他用算法来定义变异，假设 M 是算法，A 是有机体，那么变异的有机体 A’ = M(A)。适应度和进化可以用算法信息论的三个不同复杂问题定义如下。

模型 A：说出最大的自然数；

模型 B：定义快速增长的函数；

模型 C：说出康托序数（Cantor ordinal numbers）。

在模型 A 中，如果有机体能够说出一个更大的数，那么适应度就提升，有机体就得到进化。“算法信息论”中一个核心问题就是怎么用最少的资源说出最复杂的数。如果读者此时觉得有点晕，就到此为止。但如果读者此时好奇心旺盛，倒是可以看看相关参考文献。

蔡汀把自己的新理论称为“元生物学”（metabiology）。目前元生物学还比较原始，进化论里的很多概念还没有被解释，比如 Papadimitriou 关心的“性”。但有蔡汀这样的数学家关注生物学，会让我们更加放心。

参考文献指南

Holland (1975)是遗传算法的原创著作。Goldberg (1989)是教科书体例，容易上手，尽管出版日期较早，但仍有参考价值。Koza (1990)是遗传编程的原创著作，是斯坦福大学计算机系的内部技术报告，可免费获取。Koza (1992)是基于 Koza (1990)的正式出版物，后来分别在 1994 年、1999 年和 2003 年出版了第二卷、第三卷和第四卷，每卷都主打某一类应用问题。

霍兰德曾经写过几本科普读物，如 Holland (1999)和 Holland (2014)。但大科学家未必是好的科普作家，他的著作涉及的方面太多、问题太杂，给专家看看可以，但不适合完全的门外汉。另外，他的哲学观点是整体论的，而不是被大部分科学家所接受的还原论。他认为整体大于局部之和，大量的"局部"凑到一起，可以形成"涌现"（emergence）现象。从霍兰德的研究领域，如细胞自动机和遗传算法，也可看出他为什么这么说。这是他喜欢圣塔菲研究所（Santa Fe Institute）的原因，那里是研究复杂性现象的基地之一。在那儿，他可以和同道们同病相怜。霍兰德在接受采访时把圣塔菲研究所比作早年的华沙逻辑学派，一开始不被众人接受，但时间长了，自然会有影响力。米歇尔（Melanie Michell）是霍兰德的学生，她的科普著作《复杂》（*Complexity*）里面也有对细胞自动机和遗传算法的科普介绍，她的书比她老师的卖得更好。

Sutton and Barto (1998)是强化学习的原创著作，也可作为教科书，该书 2017 年出了第二版，第一版和第二版的初稿在网上可免费获取。强化学习的教科书里最爱用的 Q-learning，是 Chris Watkins 1989 年在他的剑桥博士论文里提出的。

Kubat (2015)是一本非常可读的机器学习导论，并且有中译本，最后一

章是“强化学习”。周志华的《机器学习》教科书的最后一章也是“强化学习”。Russell 和 Norvig 的人工智能经典大部头教科书，全书由 7 篇组成，“强化学习”是“学习”一篇里的最后一章。这大概说明强化学习比较新吧。

蔡汀的《证明达尔文》是科普读物，但要是想了解蔡汀思想的全貌，必须要懂他的算法信息论，那就得有基本计算理论和数学的知识。理论计算机科学家阿伦森（Scott Aaronson）曾经写过一篇非常有意思的科普文章“谁能说出更大的数”（Who Can Name the Bigger Number），这可以是算法信息论的入门。和蔡汀类似，图灵奖获得者、理论计算机科学家瓦连特（Leslie Valiant）也出过一本科普读物 *Probably Approximately Correct*，语言通俗，尤其头两章非常可读，但后面的核心内容却不易懂，因为他有个所谓由该书书名几个单词的首字母命名的 PAC“可学习理论”，那个理论需要点计算理论的知识。一个有意思的观察是几位理论计算机科学家开始关注机器学习，而且多从生物学的角度切入。瓦连特甚至说，计算机科学关注人比关注计算机更多（more about human than about computers）。

第 9 章
哲学家和人工智能

The real discovery is the one that makes me capable of stopping doing philosophy when I want to, the one that gives philosophy peace.

真的发现是能让我停下来不做哲学的发现，让哲学消停的发现。

——Wittgenstein（维特根斯坦）

哲学家不一定懂哲学，就像相声演员不一定会说相声，这是低门槛行业的通病。本章只讲哲学家和人工智能的故事，而真正的人工智能中的哲学问题散布在别处。加州大学伯克利分校的哲学系有两位人工智能的出名批评者，一位是休伯特·德雷弗斯（Hubert Dreyfus），另一位是塞尔（John Searle①）。德雷弗斯的《计算机不能干什么》和塞尔的假想实验②“中文屋”一度代表了外行反对人工智能的最强音。2016 年去世的普特南（Hilary Putnam）身兼哲学家、数学家和计算机科学家，对自动定理证明做过原创贡献，是内行。他的“缸中脑”是严肃的假想实验。

1. 德雷弗斯和《计算机不能干什么》

休伯特·德雷弗斯是美国为数不多的欧陆派哲学家之一，主攻胡塞

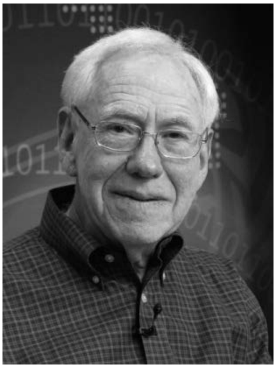
① Searle 发音更接近“瑟尔”，但“塞尔”是更广为接受的译名。

② 假想实验，thought experiment，也译为“思想实验”。

尔、海德格尔和梅洛–庞提（Merleau-Ponty），但他出名主要靠的还是批判人工智能。

休伯特·德雷弗斯是苦出身，但本科、博士都毕业于哈佛大学，刚入学时读过一段时间物理，很快就转了哲学。他还有个兄弟斯图亚特·德雷弗斯（Stuart Dreyfus），也是哈佛博士。中西部的农民家庭一家出两个哈佛博士倒是个励志的故事。斯图亚特·德雷弗斯的专业是应用数学，毕业后也去了加州大学伯克利分校，在工业工程与运筹学系任教，曾和动态规划的祖师爷贝尔曼合作，还一度研究过神经网络。兄弟俩 1988 年还联手写过一本批评专家系统的书《人定胜机》（*Mind over Machine*）。

休伯特·德雷弗斯在哈佛博士毕业前，就开始在隔壁的麻省理工学院教哲学导论，毕业后顺理成章地成了麻省理工学院哲学系助理教授。那时的麻省理工学院哲学系和语言学系还没合并。学校最热门的明星人物之一是明斯基。德雷弗斯 1961 年在他兄弟的帮助下拿到了一个在美国军方智库兰德公司夏季实习的机会，期间接触到纽厄尔和司马贺的工作。1964 年夏季他再次到兰德公司工作，写出了那篇广为流传的“炼金术与人工智能”。他在兰德工作时的老板起初看见这篇文章后非常不爽，他认为德雷弗斯的东西是糟糕的哲学，他扣着这篇东西没有立即发表。但德雷弗斯把预印本在各个圈子里散发。德雷弗斯的批评者认为那根本就不是兰德的报告，德雷弗斯不过是利用他在兰德短期工作的身份，为报告背书。9 个月之后，德雷弗斯在兰德的老板怕被别人批评在实施检查制度，最终同意以最低级别的兰德公司内部备忘录的形式印发了这篇文章。这篇文章一开始只是攻击纽厄尔和司马贺的观点，后来内容逐渐被扩充，攻击的范围也扩大，终于成为一本书，书名是《计算机不能干什么》，变成了对人工智能的全面批评。



休伯特·德雷弗斯（1929—2017）

德雷弗斯的书出来后，他马上成了人工智能学界的死敌。麻省理工学院的人工智能圈子在明斯基的影响下，根本不待见他，他在食堂吃饭时，大家都躲着他。德雷弗斯后来指控麻省理工学院的计算机教授企图阻挠他获到终身教职（tenure），因为害怕他对 AI 的批评会导致学校拿不到政府资助。他甚至考虑雇一个演员装扮成美国国防部先进研究局（DARPA）的官员和他在麻省理工学院教授俱乐部里吃饭，以此吓唬他同校的人工智能教授们。最后校长维斯纳（Jerry Wiesner）不得不干预，在咨询了邻居哈佛大学以及苏联的计算机同行后，亲自批准给德雷弗斯终身教职。维斯纳是电机工程（EE）的教授，麻省理工学院的 EE 和计算机至今还是一个系，维斯纳在计算机教授间颇有威信，著名的 Media Lab 就是在他手里建立的。苏联那时正批控制论，德雷弗斯的调子倒是合拍。后来 DARPA 传唤德雷弗斯到首都华盛顿听取他的反动言论，再后来，还真有了 AI 的核冬天，而那期间麻省理工学院人工智能实验室的政府资助也确实大幅减少。

1992 年《计算机不能干什么》出版 20 周年纪念时，德雷弗斯新写了个序，新瓶装旧酒，书又被重新出版了一遍，正文内容没什么变化，只是书名被改为《计算机仍然不能干什么》。这本书的副书名随版本不同，时有变化，或“人工智能的极限”，或“人工理性批判”（A Critique of Artificial

Reason），很明显，这是抄自康德的《纯粹理性批判》。人工的当然不纯粹，一个假酒一个茅台，不同的是：康德初衷是建设，而德雷弗斯则是大批判。更有意思的是，新版的出版社竟然变成了麻省理工出版社。麻省理工出版社毕竟不是早年结下梁子的麻省理工学院，此时的人工智能圈子应该也更有自信了吧。

德雷弗斯从以下四个层面批评人工智能。

不同层面	人工智能假设	德雷弗斯的反驳
生物层面	麦卡洛克–皮茨的神经元是二元的，像布尔电路	人脑是模拟的
心理学层面	纽厄尔–司马贺的信息处理和规则	常识和背景无法用规则表示
认识论层面	麦卡锡：所有知识都可以形式化	人的知识不是形式化的。可以用微分方程描述星体运动，不意味着星体在求解微分方程
本体论层面	世界由事实构成，方法论是还原论	人是人，物是物。物理的东西是还原论。人需要现象学

德雷弗斯的批评随着人工智能的进展也与时俱进，但可以总结为多少有些互相矛盾的两句话：其一，所有人工智能研究者搞的东西都是哲学家玩剩下的，例如霍布斯的“推理即计算”（reasoning is calculation）、笛卡尔的心理表示、莱布尼茨的“广义计算”（universal characteristic）；其二，有好多人能干的事，现在的计算机干不了。德雷弗斯甚至小肚鸡肠地推断明斯基提出框架概念肯定受到明斯基的一个学生的影响，而那个学生上过德雷弗斯的现象学的课。

纽厄尔和司马贺 1957 年曾做过一个乐观的预测：十年内计算机下棋能赢人，十年内计算机将能证明人还没有证明的定理。这成了德雷弗斯的把柄，也是他最有力的武器。每次计算机下棋有些进展，德雷弗斯都会跳

出来说："那还是赢不了人啊。"其实 1966 年德雷弗斯就曾和当时最厉害的下棋程序 MacHack 对弈过一局，并且输给了 MacHack，但他说赢我不算赢。他不得不再次澄清他原来的论断是计算机当时不能赢人类棋手，并不是永远不能赢人类棋手，这不是废话嘛。美国计算机学会人工智能组的出版物 *SIGART Bulletin* 引用了明斯基的战友佩珀特（Seymour Papert）以维特根斯坦的格式写的笔记：

1.5 计算机不能下棋。

1.5.1 德雷弗斯也不能下棋。

1965 年司马贺再度预言：他的原定目标可以在 20 年内可以实现，结果到了 1985 年，自然为德雷弗斯徒增笑料。直到 1996 年马库恩（McCune）的定理证明程序 EQP 证明了罗宾斯猜想，1997 年 IBM"深蓝"战胜象棋世界冠军卡斯帕罗夫。常被德雷弗斯讽刺的另一个领域——几何定理证明——在 1978 年吴文俊的工作之后也渐趋成熟。不知道如果现在德雷弗斯再写一版那本书，书名该改成啥，《计算机就是不能干》？不过人工智能的乐观派也该学到点教训：老老实实干活，没事别瞎吹。

1986 年，麻省理工学院人工智能实验室的老大温斯顿（明斯基的学生）邀请已经在加州大学伯克利分校任教的德雷弗斯回来做了个讲座，题目居然是"为什么 AI 从业者应该读《存在与时间》？"《存在与时间》是德国哲学家海德格尔的成名作。海德格尔是德雷弗斯偏爱的哲学家，素不为英美哲学家所喜。2008 年德雷弗斯还写过篇文章"为什么海德格尔派的 AI 失败了，为什么需要更多的海德格尔"，大意是人工智能中的重要思想都来自海德格尔，而正是因为贯彻海德格尔思想的不彻底导致了人工智能的失败，为了成功，我们需要更多的海德格尔。换句话说，成功是因为听从海德格尔的教导，失败是因为没听从他的教导。德雷弗斯文中似乎把人

工智能没有成功的主要原因都归结为“框架问题”，并提出“框架问题”是不能通过符号派的表示手段来解决的，唯一手段是利用神经网络，而他所谓的神经网络研究也被海德格尔早就料事如神地在《存在与时间》里想到了。德雷弗斯认为明斯基一票人不懂海德格尔，所以导致知识表示的框架问题。这听起来真有点儿像仁波切们说他们早就料到了量子力学一样。

德雷弗斯声称明斯基的“框架”式知识表示的根源是海德格尔的犹太人师傅胡塞尔（Edmund Husserl）。胡塞尔的现象学就是知识表示系统，只是胡塞尔在他 75 岁（1934 年）时认识到知识表示是死路，而明斯基在 1988 年还没认识到。德雷弗斯这么说的根据是明斯基的框架系统有个顶层概念，而胡塞尔也有个东西叫“神圣不可侵犯的本源”（Inviolably the same）。如果认真读过胡塞尔，就知道胡塞尔现象学和知识表示没啥关系，只是有修辞性的比喻，而非实质性的相似。德雷弗斯大概知道科学家对胡塞尔和海德格尔的晦涩语言伪装的深刻不感兴趣。

哲学家有两类，一类是深刻的，一类是混饭的。罗素和弗里格是深刻的，没有他们，就不会有数理逻辑，也就不会有哥德尔、丘奇、图灵，以及后来的计算机科学。但没有现代的欧陆哲学，世界不过省了些粮食而已。没有胡塞尔和海德格尔，明斯基照样会想出“框架”，从而催生后来的“面向目标的程序设计”方法论。所谓“顶层”概念就是 Java 程序设计语言里的 Object，或者知识图谱 DBpedia 里的 Thing。按照德雷弗斯们的说法，哲学系是不是应该要求读现象学的博士必须熟练掌握一门面向对象的程序设计语言？

在 20 世纪 80 年代末期，神经网络研究复兴之后，德雷弗斯对人工智能的全面批评也缩小为对符号派的专门攻击。他和他的兄弟斯图亚特·德雷弗斯一起撰文写书。斯图亚特虽然是运筹学专家，但一直都在做神经网

络的研究，甚至号称发明了“反向传播”（back-propagation）的原始概念。考虑到这一点，他们的攻击不免有报复和落井下石之嫌。

德雷弗斯曾经引用梅洛–庞提批判人工智能：人脑是和环境直接交流的，而不是通过表示（representation）。德雷弗斯曾经按照海德格尔的思路创造了一个词：“随手拈来”（readiness-to-hands），也就是说直接性是不经过表示的。可以把这算作对符号派的朦胧批判吧。但这种批评有点像算命常用的两头堵，因为你无论怎么直接地接近环境，他都可以说这和人脑不同。不知道波普尔会怎么评价海德格尔，或者德雷弗斯的这种解读。很可惜德雷弗斯不知道深度学习，对多层的解释恰恰是因为需要表示。造飞机不需要按照鸟的结构，飞机的翅膀不会动。飞机的表示是空气动力学。飞机能飞吗？

明斯基和佩珀特的学生维诺格拉德（Terry Winograd）加入德雷弗斯和塞尔的批评阵营，确实给他们增色不少。维诺格拉德早期在麻省理工学院的研究课题“积木世界”是自然语言处理的经典工作。在碰到问题时，维诺格拉德放弃了原来的研究方向。他联手德雷弗斯和塞尔一道批评麻省理工学院的学生只能将课题限制在“微世界”而不是“实际世界”。但罗马不是一天建成的，任何一门科学一开始不都是从小处着手吗？所谓“微世界”就是维诺格拉德的积木世界。其实积木世界从某种意义上体现了维特根斯坦后期《哲学研究》中的思路，语义就是与环境的交互。用维特根斯坦的例子，师傅对徒弟说“递给我一块砖”，如果徒弟真递过来，那就是听懂了。

德雷弗斯能够如此长久地混迹人工智能的江湖，有两个原因：其一，人工智能内部一直就是两大派不和，而且派中有派，很少有过学科的发展像人工智能这样起伏跌宕，苍蝇不叮无缝的蛋；其二，相当一批人工智能

工作者有哲学情怀。2001 年第一次互联网泡沫崩裂时，德雷弗斯又出了本大批判互联网的书《关于互联网》，风格一如既往。他的出发点是互联网的隐私和媒体的责任感。这倒更符合他的训练和智力。

2. 塞尔和中文屋

塞尔是德雷弗斯之后又一位批评人工智能的干将，但他主要以英美哲学立身。他是地道的美国人，一开始就读威斯康辛大学麦迪逊分校，但三年级时获得罗德奖学金（Rhodes Scholarship），去了英国，结果本科硕士和博士三个学位都是在牛津拿的。他在牛津时跟随日常语言学派的领袖奥斯丁（John Langshaw Austin），回美国后马上就到加州大学伯克利分校教书，出名很早。塞尔晚年还被中国的清华大学和华东师大聘为名誉教授。2017 年初，84 岁高龄的塞尔被他的一名博士后以“性侵害”罪名告到法庭，这个罪名比“性骚扰”要厉害一级。而据他的同事说他素有这个毛病，过去就有多名学生和他发生性关系以换取金钱和分数的好处。伯克利校方既然知道他的不当行为却不加处置，为此在案件中也被连带。他原本在 2017 春季要教的“心智哲学”的课也被取消了。



塞尔（1932— ）

言归正传，1980 年塞尔在《行为与脑科学》杂志上发表了“心灵、大脑和程序”（Minds, Brains and Programs）一文。文中的一个思想实验“中文屋”马上成为最喜欢被引用的假想实验之一。曾有人批评《行为与脑科学》杂志不严谨，更像哲学杂志。但说实在的，这杂志经常登些好看的文章并屡次挑起事端。当年彭罗斯的《皇帝新脑》（*Emperor's New Mind*）一书出来后颇引争议，《行为与脑科学》就搞了一期争辩专刊，正方反方吵得不亦乐乎，各方都抬出了自己的大佬，无论从吸引眼球还是严肃讨论的角度看，这都是成功的。

所谓“中文屋”思想实验是这样的：假设有个只懂英文不懂中文的人（塞尔的第一人称“我”）被锁在一个房间里，屋里只给“我”留了一本手册或一个计算机程序，这个手册或程序教“我”在收到中文信息时如何用中文应对。屋外的人用中文问问题，屋里的“我”依靠程序用中文回答问题，沟通方式是递纸条。塞尔的问题是：假设屋外的人不能区分屋里的人是不是母语为中文，那么屋里的“我”是不是就算懂中文？塞尔自己认为“我”不懂中文。很明显，这个场景源自图灵测试，只不过图灵测试的环境是英文，而中文屋里既有中文又有英文。

塞尔的文章出来后，引起轰动。其实轰动的原因很简单：谈论这种玩意儿没什么门槛，谁都可以说三道四：哲学家、科学家，以及各种媒体人。塞尔毕竟是老练的哲学家，已经预测大家会质疑他的论断，他在文尾也设想了各种回答。中文屋的第一个问题是，我们只是算屋里人理解中文呢，还是屋子加人作为一个系统理解中文。塞尔的论断是屋里人即使查遍手册，顶多算是理解语法，而不算理解语义。我们可以问塞尔这样的问题：一个戴眼镜的人能算看见东西吗？一个耳聋的人通过换上人工耳蜗重获听觉后算是能听见吗？一个坐飞机的人算能飞吗？如果对这些问题的答

案都是“算”，那中文屋作为一个系统为什么不算理解中文呢？

塞尔认为必须内化（换句话说：手册必须变成人身的一部分）才能算懂中文，那么内化到什么程度才能算呢？爱因斯坦说“我的笔加上我要比我自己聪明”，笔算不算外化？原来纸笔时代的拼写错误，现在用任何文本编辑软件，如微软 Word，都可自动纠错，这算内化吗？内化就是一点外部工具都不能借助吗？内化是完全的物理隐藏，还是只是个反应时间问题？在一开始查手册时，反应时间必定很慢，但熟能生巧之后，查手册变成下意识的动作，那算内化吗？即使中文是母语的人也免不了查手册啊。我猜对塞尔来说，可能人工耳蜗算是内化，飞机肯定是外化，而眼镜则是可算可不算。

内化和辅助工具的大小也有关系。目前语音识别被认为是已经解决的问题，机器的识别率已经达到或超过人类。两个讲不同语言的人可以借助语音识别工具沟通。如果语音识别工具是桌面电脑，我们可能不会认为对话中的两个人理解了对方的语言。但如果这个语音识别工具可以微型化，直接内化到耳朵里，那算不算理解？同理，可能中文屋让人不能接受的原因是因为它的体积。就像丹尼特所说“植物是很慢的动物”，我这里可以再加一条“植物是很大的动物”。如果我们可以把基本功能做得很小很快，那么“内化”就是很便宜的功能。

在塞尔的术语里，理解或意识等同于一个抽象的哲学观念“意向”（intention）或“意向性”（intentionality）。屋里的人并没有“意向”，所以“我”没有理解中文。在塞尔的论辩里，时而意向性是人特有的性质，时而意向性是不可检测到的东西。

塞尔认为他不是反人工智能，他只是反“强人工智能”，但在“强人工智能”和“弱人工智能”之间并没有质的区别，只有量的渐变。中文屋

测试的不是屋中的“我”而是屋中的程序。如果那本神奇的手册或者程序已经通过图灵测试，那程序就是一个机器翻译的神器。这本身就是强人工智能了。而且那程序已经有语义功能了。假设游戏不是中文翻译，而是下棋，那“我”算不算会下棋？断言中文屋是不是有智能，就像断言 AlphaGo 会不会下围棋一样，要看应用场景。一个数学家离开了数学也不一定就是聪明的。

塞尔的第二个准备好的答辩就是所谓“机器人”反驳。如果那本手册或者那个程序那么厉害，如果把它放到一个机器人里，那么这个机器人就可以做很多人可以做的事情，那么它是不是就算能理解了呢？塞尔的答辩是这恰恰说明单纯的形式化符号操纵是没有理解力的。

维特根斯坦后期哲学《哲学研究》中有个例子：泥瓦匠要徒弟把砖头递给他。如果徒弟把砖头递过来了，那么徒弟就是懂了。理解是一个社会现象，而不是一个独立的哲学概念，或个人行为。按此，语义也是社会的。翻译只是心心交互？翻译需要涉及外部世界吗？维特根斯坦说语义就是语言的使用。语言的使用也必涉及心物交互。从这个角度看，也许关于中文屋的讨论可以更有建设性。

3. 普特南和缸中脑

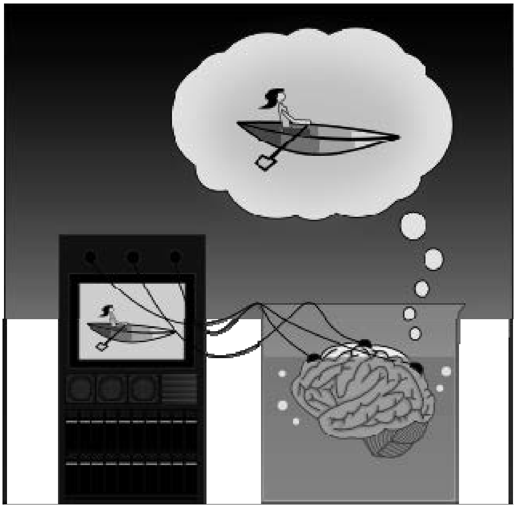
普特南灿烂的学术生涯覆盖了数学、计算机科学和哲学。他虽是哲学出身，但他也是解决希尔伯特第十问题的主要推手之一，他和逻辑学家戴维斯长期合作研究机器定理证明，是这个领域的开拓者之一。他的哲学研究也涉猎广泛，并且立场常常变来变去。他的变化和德雷弗斯的浅薄机会主义不同，普特南是高处不胜寒，自己换着法儿和自己辩论。他在 1960

年就写过一篇“心和机器”（Minds and Machines），定义了计算主义（Computationalism）和功能主义（Functionalism）。普特南指出同样的软件可以在不同的硬件上运行，软件的功能和硬件的实现可以分开。到 20 世纪 80 年代初，普特南又变成计算主义和功能主义的批判者。

1981 年普特南出版了《理性、真理与历史》（*Reason, Truth, and History*）一书，该书的开篇就给出了“缸中脑”的假想实验：

“一个人（想象一下那是您自己）被邪恶科学家施行了手术，他的大脑（您的大脑）被从身体中取出，放入一个缸中，缸里盛有维持脑存活需要的营养液。脑的神经末梢和一台超级计算机[①]相连，这台计算机使大脑的主人保持一切完好的幻觉。对于他来说，似乎人、物体、天空还都存在，但实际上，此人（您）体验到的一切都是计算机传输到神经末梢的电子脉冲的结果。这台计算机非常聪明，此人要是抬起手，计算机发出的反馈能让他“看到”并“感到”手正在抬起。不仅如此，邪恶科学家还可通过改变程序使受害者“经验到”（即幻觉到）邪恶科学家所希望的任何情景或环境。他还可以消除这次脑手术的痕迹，从而使受害者觉得自己一直是处于这种环境。受害者甚至还会以为他正坐着读书，读的就是这样一个有趣但荒诞的故事：一个邪恶科学家把人脑从人体中取出放到一个有营养液的缸中。神经末梢连到一台超级计算机，它使这个大脑的主人具有如此幻觉……”

① 原文是 super-scientific computer。



缸中脑假想实验

普特南更进一步设想，假设所有的感觉器官都泡在缸里，而外面的世界就是一台大自动机。美国科幻电影《黑客帝国》（*Matrix*）、《盗梦空间》（*Inception*）等都受“缸中脑”思想实验的启发。

普特南发明了一种新的图灵测试，他称之为“图灵指称测试”（Turing Test for Reference），测试的方法和图灵测试一样，通过传递打印纸条，来判断是否机器能像人那样指称外部世界的客体。普特南的结论是机器不能像人那样指称。图灵指称测试实际上已经很接近塞尔的中文屋。尽管如此，普特南一直对“意向性”这样模糊不清的概念不爽。

普特南进一步把缸中脑和图灵指称测试做了对比。图灵指称测试的底线是语言，任何超越语言的东西，图灵测试没法回答，例如，如果给机器一张丘吉尔的照片，机器会不知所云。但缸中脑的底线是神经末梢，给缸中脑一张照片，缸中脑知道如何反应，缸中脑毕竟也是脑啊。但问题是缸中脑知道如何与外部世界做对应吗？泡在缸中的人脑，如何知道自己是颅中脑，还是缸中脑？人工智能的基本问题是可否造一台机器能有智能，

“缸中脑”中的机器则起了另一种作用：人脑是否能确定外在的世界是直接实在还是间接实在。

普特南曾经正面地批评过人工智能：人工智能并没干什么哲学家不能干的事。但这要看是什么人工智能的学者和什么哲学家。如果是普特南这样的哲学家，计算机科学家必须认真倾听，因为普特南自己就是一位富有成就的计算机科学家，他也是人工智能一些分支的开拓者，计算机科学家大概会更加认为他是自己人而不是他者。他在塔夫茨大学的邻居丹尼特（Daniel Dennett）则会对人工智能采取更加同情的态度。他认为即使人工智能没有解决什么实际问题，但从更深的层次提出问题本身就是进步。哲学家总不会因为生物学家没有解答什么是生命，就怪罪生物学没有取得进展吧。没有人工智能，有些问题根本就提不出来。普特南和丹尼特都算是建设性的意见，区别是一个把人工智能当成自己的用户，而另一个把自己当成人工智能的用户。

4. 给哲学家一点忠告

曾经有一个教条：哲学指导科学。这种观点为大多数科学家所不齿，费曼、惠勒和杨振宁等物理学家都曾撰文批驳。但这恰是德雷弗斯的天真立场。维特根斯坦曾经有言：哲学家的工作应该是一直给人提醒（assembling reminders），而不是指导。德雷弗斯最早对人工智能的批评其实很简单：还不能做什么。比如最早他说下棋还不能做，当计算机科学家证明他不对时，他又说还有什么仍然不能做。如果把这些都当作维特根斯坦所谓的“提醒”，倒也不错。但比较让计算机科学家愤慨的是，他会常常会以教导的语气说，你们应该照着海德格尔说的做。

在文明初期，哲学家掌握所有的学问，哲学就是学问的代名词，说哲学家指点科学倒也不算错。但科学进步的过程就是与哲学渐行渐远的过程，当下的科学已经和哲学关系不大，一战后的欧陆哲学已经和科学彻底无缘。偏重科学和逻辑的英美分析哲学也挡不住哲学的颓势，最后一个从哲学中脱离的硬学问是逻辑，目前最好的逻辑学家都在数学系和计算机系，哲学已经空洞化。那些非逻辑学出身的哲学家存在的一个普遍问题是压根就没见过硬的、复杂的问题。对一个不太出名的哲学家的谬论，最好的应对办法是把他交给比他数学稍微好一点的同行。但如果碰到出名的哲学家，我们只好直接迎战，否则他的谬见会影响智力还不如他的媒体人，从而被散布得更远。

彭罗斯其实也看不起塞尔，他在《皇帝新脑》书里评论塞尔时说塞尔被计算机科学家误导了，而计算机科学家又被物理学家误导了，这明显表示了某种学科的智力层级的歧视。彭罗斯把塞尔的几种辩解轻易批倒，但他还是喜欢塞尔的结论：强人工智能不靠谱。颇有否定之否定或敌人的敌人的意思——彭罗斯自己作为物理学家不认同强 AI，强 AI 多出自计算机科学家，而哲学家塞尔又是反对强 AI 的。这是为彭罗斯引出自己的理论铺路的：这就是人脑实际上有量子效应的。我们也由此可以看出科学家和文科生有时是一笔乱账地互相寻求支持，科学家内心知道哲学家之不靠谱，而哲学家有时特别需要科学家的背书，尽管科学家内心知道哲学家对自己工作的陌生和胡乱引用。

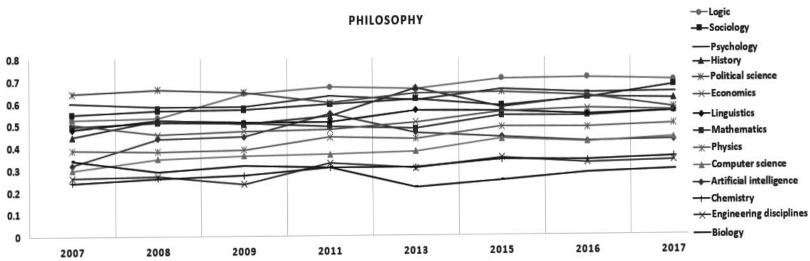
德雷弗斯批评人工智能太重分析，而不够综合，因而提倡现象学。生物学家、诺贝尔奖金获得者埃德尔曼（Gerald Edelman）和他的学生里克（George Reeke）则说人工智能太过综合而不够分析。他们在合作的一篇文章“实在的大脑和人造的智能”（Real Brain and Artificial Intelligence）开篇中

就讽刺亚里士多德——亚里士多德的《动物学》里陈述女人的牙齿数目比男人少，亚里士多德从来也不看亚里士多德夫人的嘴。埃德尔曼希望计算机科学家应该等神经科学家了解了大脑的生物学之后再谈论人工智能，或者干脆加入神经科学家的队伍一起先研究大脑。但计算机科学家恐怕没那个耐心。我们造飞机并不需要知道鸟是怎么飞的。我们享受飞机的远程旅行，也不需要懂空气动力学。我这里并非在为功能主义辩护。其实，学科体系历来如此，底层的学科说上层学科不够分析，物理学家对化学家如是说，化学家对生物学家如是说，生物学家对心理学家如是说。而哲学和人工智能恰恰可以和哪个学科都能挂上，分析和综合就看不清了。

如果真认为海德格尔有用，就应该像弗里格和罗素清理逻辑那样，把这些东西整理成可以交流的形式。也许哲学家真怕他们惯用的冷僻词汇被翻译成通俗易懂的语言。当代哲学，尤其是欧陆哲学，就像韩国整容术，乍一看唬人，其实遗传不了。

彭罗斯曾经这样谈到机器的情感和道德：如果你买一台计算机，它是有情感的，那么我们就有道德问题，因为计算机的意愿可能被违反，并可能会被当作奴隶。我们首先必须说道德是一个社会问题，也就是说当一个社会只有一个个体（无论是人还是计算机）时，是不存在道德问题的。

哲学家很喜欢对人工智能说三道四，原因可能是人工智能关心的问题，例如意识、生命、思维、自由意志等概念，都是哲学家自认固有的地盘。但如果我们用谷歌距离函数计算维基百科中所有学科的距离和关联度，我们会发现人工智能和哲学的距离并没有想象的那么近，也没有想象的那么依赖哲学。



维基百科中哲学和其他学科的关联度

创造性非虚构（creative nonfiction），例如传记，也算文学的一类，普利策奖就为此有专奖。当下的大部分哲学家都是从事创造性写作但又没有创造能力的人，他们是没有才华的文学家。计算理论家阿伦森（Scott Aaronson）曾写过一篇文章“为什么哲学家应该学点计算复杂性？”，婉转地表达了对那些喜欢随意地对计算理论说三道四的哲学家的期望。学点计算理论，少说外行话，至少能有助于哲学家了解科学家在谈论什么问题。计算理论的源头可追溯到罗素、维特根斯坦和哥德尔，他们都有哲学背景。甚至图灵也有哲学情怀，他甚至被哲学家兼传记作家蒙克（Ray Monk）称为与柏拉图、罗素并列的十大哲学家。我一直很好奇为什么对人工智能感兴趣的欧陆哲学家比他们的分析哲学对头更多。

丹尼特曾说哲学家喜欢假想实验。其实从某种意义上说，整个人工智能就是个大的假想实验。只不过哲学家用纸和笔，而计算机科学家用计算机硬件和软件。本质是一样的。不同的是哲学家从不为假想实验的结果所苦恼，反而会时不时洋洋自得；而计算机科学家则偶尔会被他们取得的成果所惊到。崇尚科学的英美分析哲学家和欧陆哲学家分歧已久。英国哲学家克里切利（Simon Critchley）曾经写过一本毒舌的《哲学家死亡录》，里面记录了有史以来哲学家的各种死法，他的结论是分析哲学家的死大多是无趣的，而欧陆哲学家的死则多彩缤纷。随着大数据手段的成熟（套用

德雷弗斯的话说，所谓成熟就是“随手拈来”），也许哲学家又重新有机会对科学的全局观有所洞见。到底应该是哲学家向科学家学习，还是科学家向哲学家学习？让科学家写一本《哲学家不能干什么》要远比哲学家写《科学家不能干什么》容易得多。

第 10 章
人是机器吗？——人工智能的计算理论基础

...humans are nothing but meat machines that carry
a computer in their head.
……人不过就是脑袋上顶了个计算机的肉机器而已。
——Marvin Minsky（明斯基）

那些认为人是机器的，其道理很简单：人也是由各种物理化学机制构成的，当然是机器了。早有法国哲学家美特里，现有 DNA 双螺旋结构发现者克里克，都持这种观点。克里克认为在不远的将来，生命可以在试管中合成。而那些认为人不是机器的，其论据是人有很多功能，目前机器无法完成，尤其是那个叫“灵魂”的神奇东西。“灵魂”被苏格拉底用来论证永生的可能性，算是他被执行死刑之前给自己的宽心丸。

“人是机器吗”这个问题的另一个角度是比喻性的，计算机科学家蔡汀（Gregory Chaitin）在半科普半学术的小册子《证明达尔文》中把物理学和生物学做了对比，他给物理学贴的标签是“死的，刚性的，封闭的，机械的”，而把“活的，可塑的，开放的，创造性的”这些美好的词汇都赋予了生物学。物理学是硬件，生物学是软件。从这个意义上说，生物学和物理学的对立就是人和机器的对立。

计算机科学起源于图灵 1936 年那篇无论怎么夸赞都不过分的文章“论可计算的数”，这是人类文明最重要的成果之一。图灵在这篇文章中定

义了后来被他的导师丘奇称为“图灵机”的计算装置：一条无穷长的纸带，一个读写头在一个控制装置的控制下在纸带上方左移右移，读取纸带上的内容并在纸带上写 0 或 1。图灵的初衷是让他的机器模仿人类计算者（computer），那时 computer 的涵义是人类计算者，而机器计算者是 computing machinery。

“人是机器吗”这个问题有很多同源的古老哲学问题，例如，“心–脑”（mind-brain）和“心–身”（mind-body）。还有很多相关问题，例如，自由意志和自我意识。如果人是机器，那么他们是模拟机器还是数字机器呢？

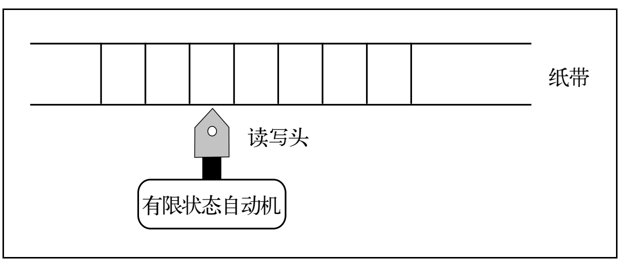
按照冯诺伊曼的说法，神经系统的本质是数字的，尽管构成神经系统的化学和生物过程的描述可能是模拟的。现代物理学的一个假设是整个宇宙都是离散的，也即数字的。如果机器是数字的，那么图灵机就是简单又有力的模型。做人工智能绕不过去图灵机和在其上建立的整个计算机理论。人工智能符号派的基础之一是所谓“物理符号假设”，这个假设要求计算装置必须是数字的，或者说变量必须是离散的。我们有一些直觉：离散的量是区分和定量的基础，手指的数目是离散的，由此计数也是离散的，语言也是离散的。费曼就曾说世界是数字的。对于离散的量，二进制就足够了。朴素唯物主义认为世界是连续可分的，从某种宏观的意义上说，朴素唯物主义是经典物理的思想基础，当然像凯恩斯那样把牛顿当作最后一个魔术师而不是第一个科学家，也没什么错。历史问题有点像海岸线问题，尺度不同则结论也不同。新的量子物理认为世界是离散的、有限的。如果从物理学角度认可离散，那么化学和生物的角度也必然是离散。连续变量是离散变量的一种数学近似。

按照这个思路，“人是机器吗”这个问题就被归约到“人是计算机器吗”，又进而被归约到“人是数字计算机吗”。如果我们把“智能”当作人

类特有的性质，那么“人是机器吗”这个问题就等价于“机器有智能吗”。

1. 丘奇–图灵论题：为什么图灵机是最重要的发明？

在人类发明的所有计算装置中，图灵机是直觉上最简单最可靠的。图灵机有三个组成部分：一条无穷长的纸带，上面有无穷多个格子，每个格子里可以写 0 或 1；一个可以移动的读写头，每次可向当前指向的格子写入 0 或 1；一个有限状态自动机，可以根据自身的状态，以及当前纸带上的格子是 0 还是 1，指示读写头向左或向右移动一个格子，或向当前的格子写入内容。就是这样一个简单得不能再简单的装置，被图灵证明和丘奇的 λ 演算是等价的，而 λ 演算又被证明和哥德尔的递归函数是等价的。图灵机的强大能力恰恰是因为它的简单。哥德尔曾经对自己的递归函数是不是最广义的计算装置不是很有自信，但在了解了图灵机的构造后，他完全信服了，但哥德尔还是不认为人是机器。



图灵机

在计算理论里，有一个著名的丘奇–图灵论题（Church-Turing Thesis）：所有功能足够强的计算装置的计算能力都等价于图灵机。这是一个观察，而不是定理。图灵、丘奇、克里尼等人证明了当时所有数学家和逻辑学家想出的各种计算装置（例如递归函数、λ 演算、Post 系统、图灵机等）都

可以互相模拟。

丘奇（Alonzo Church）是土生土长的美国数学家，他是和塔尔斯基、蒯因同辈的人。他的老师是美国数学家奥斯瓦尔德·维布伦（Oswald Veblen）。正是在维布伦手下，普林斯顿大学的数学系成为一流，维布伦也是普林斯顿高等研究院的创始者之一，还是这个研究院的第一个教授。丘奇在短期游学欧洲之后，就一直在普林斯顿大学的数学系和哲学系教书。他 1967 年从普林斯顿大学退休后又到加州大学洛杉矶分校发挥余热。图灵正是在他的指导下得了普林斯顿大学的博士学位，那时英国人得美国学位又回到英国教书的真是不多。在丘奇不长的博士生名单中可以见到戴维斯、克里尼、罗杰斯，以及因为有限自动机而得图灵奖的拉宾和斯考特，等等。“图灵机”的说法也源自丘奇，他 1937 年在为《符号逻辑杂志》撰写图灵“可计算的数”文章的评论里首次用了图灵的名字。



丘奇（1903—1995）

λ 演算的来源也是件有意思的事。怀特海和罗素的大部头《数学原理》里面有 $\hat{x}$（x^*x）的记法，表示括号里的 x 是变量。$\hat{x}$ 发音为 x-hat，就是 x 戴个帽子的意思。丘奇仿效了罗素的记法，但排版师傅不知道怎么排，

就把大写的希腊字母Λ（发音“朗目达”）来表示，后来又慢慢演变成小写希腊字母的λ。因为丘奇的整个演算系统都是基于这个帽子，于是λ就成了丘奇无名函数的代名词。想到图灵的机器被丘奇命名为图灵机，函数程序设计中的 currying（柯里转换，或柯里化），也是基于逻辑学家柯里（Haskell Curry）的名字，这个本应称作丘奇演算的东西却被歪打不正着地叫成λ演算。如果它被称为丘奇演算的话，是不是丘奇的历史地位会更加高大一些，毕竟 λ 演算的发表日期要早一些，也是他最早意识到图灵机的重要性。随着历史的进步，丘奇被掩盖在了图灵的光芒之下。

丘奇–图灵论题没法被严格地证明，因为无法知道还会不会有新的计算装置被发明，只能说基于目前的观察，这是真的。这是归纳，不是演绎，所以这不是数学定理，更像是物理定律。但那么多最聪明的人想出来的玩意都是等价的，确实给人很多信心：这些就是最强的计算装置。20 世纪 50 年代，乔姆斯基发明生成语法后，很容易就证明了乔姆斯基 0 型语法等价于 Post 系统，那自然也等价于图灵机了。天才数学家冯诺伊曼发明的细胞自动机后来被天才物理学家沃尔弗拉姆（Stephen Wolfram）发扬光大，也被证明和图灵机等价。物理学家多依奇（David Deutsch）认为丘奇–图灵论题应该像能量守恒定律一样被接受。

我们可以用图灵机来模拟任何一个程序，假设现代的输入输出设备可以接到一个图灵机上，我们甚至可以用图灵机实现一个操作系统的模拟器，那么我们就可以在纸带上跑 Windows 或 iOS。这其实就是现代虚拟机的概念。虚拟机的一个后现代非专业的中文译名叫“透明计算”。有好事者真用纸带和机械零件实现了一个图灵机，但甭管多“透明”，在上面跑 Windows 恐怕还是不太现实。



可运行的纸带版图灵机

图灵在发明图灵机时，还定义了 Universal Turing Machine，简称 UTM，译为“广义图灵机”或“万能图灵机”或“通用图灵机”。UTM 的核心思想就是一个图灵机的执行过程也可被编码成数据，放到纸带上，这样一个图灵机就可以把被编码的图灵机指令读出来，一步一步地执行，从而模仿这个特定图灵机的行为。这样，这台能模仿其他图灵机的图灵机就成了万能图灵机。这是一个很深刻的思想，现在的软件产业都得益于此：被编码的图灵机就是软件。后来冯诺伊曼设计的计算机被人称为冯诺伊曼架构，其最核心的思想就是存储程序（Stored Program）。这个思想其实就是来自万能图灵机：被编码的图灵机就是存储的程序。对冯诺伊曼架构的各种误解主要发源于这样一批人：他们对计算理论和历史零知半解，却又喜欢道听途说。冯诺伊曼架构的真正原创是随机寻址，但这是个实际考量，而不是理论设计。冯诺伊曼把计算机的所有原创思想的功劳都给了图灵，并批评那些对图灵机实际意义缺乏认识的人。

有了图灵机，我们就很容易把原来是纯逻辑或纯数学的东西（例如递归函数和 λ 演算等）和物理世界联系起来了，函数成了纸带和读写头。如果按照传统哲学的说法：数学唯心，理性；物理唯物，经验，那么计算

机科学就是唯物和唯心的桥梁，司马贺把这类东西统统叫作“人工科学”。物理学家费曼说计算机科学既不是科学，也不是数学，有点像工程。无独有偶，算法大师高德纳（Donald Knuth）把科学分为三种：物理科学（如物理、化学、生物等），数学，计算机科学。他把后者称为“非自然科学”（unnatural science）。高老说他读知识史时，总觉得有些先哲，如果早有计算机系的话，肯定是计算机系教授，他没说是谁，但我猜他的名单上应该有莱布尼茨和罗素。另外他说计算机科学有别于数学和物理的地方在于常常在不同抽象层次间跳来跳去，而数学只抽象，物理太具象。他估算每 50 个人里会有一个人有计算机科学的思维方式。对高老来说，最高的抽象当然是上帝了，他是科学家中为数不多的有神论者，是虔诚的路德教派信徒，他认为生物学比计算机科学要深得多。

2. 相似性原则：另一个重要但不太被提及的计算理论思想

丘奇–图灵论题宣称所有足够强的计算装置都可以互相模拟。但模拟的成本如何呢？会不会存在一种计算装置比其他计算装置更有效呢？这就要说到计算复杂性理论了。

有些问题，凭直觉我们也能想出算法，例如排序。没有计算机科学的专业训练，最容易想到的算法是冒泡，就是在数组里找到最大的元素，把它挪到最前面，然后找到次大的元素，挪到最大的后面，以此类推。如果这个数组长度是 n，那么冒泡算法需要大致 n^2 个步骤。学过计算机科学的人都知道冒泡排序不是最有效的排序方法，有各种 $n\log(n)$时间的排序算法。在数据量很大时，这些排序算法的效率要远高于冒泡法。算法的时间

和空间效率的衡量是计算机科学的核心问题之一。那么怎样才算是高效率算法呢？一般的看法是，运行时间和空间随着问题规模扩大而变得很大的就不是高效的算法。

关于什么是大什么是小，其实是很难定义的。看看王浩悖论：1 是小的数（small number），如果 n 是小的数，那么 $n+1$ 也是小的数。是不是按照数学归纳法可以得出结论：所有自然数都是小的数呢？目前在计算复杂性理论里，大家公认多项式是小的，而指数是大的。算法的运行时间或空间随着问题规模呈多项式函数增长的就算是高效算法，因为多项式的增长相对于指数增长是更加可控的。计算复杂性是指算法的最坏情况。但有时最坏情况发生的概率很低，快速排序大部分情况要快于其他排序算法，但快速排序的最坏时间复杂性是 n^2。另一个例子是线性规划的单纯型算法，最坏情况是指数的，但日常情况表现很好。这就引出了平均复杂性分析。

有些问题内在就是很难的，不大可能会找到有效算法，例如旅行商问题（TSP Traveling Salesman Problem，简称 TSP）。TSP 在非确定图灵机上可以在多项式时间内求解。非确定性图灵机多项式时间可解的问题简称 NP，确定性图灵机上多项式时间的问题简称 P。最难的 NP 问题被称为 NP 完全的。第一个 NP 完全的问题是库克在 1971 年受王浩的启发在研究机器定理证明时得到的，这个问题就是命题逻辑的可满足性问题，简称 SAT。后来加州大学伯克利分校的数学家卡普（Richard Karp）证明了一系列问题都可以和 SAT 在多项式时间内互相归约。这一类问题就都是 NP 完全的。非确定性图灵机多项式可解的问题可以在确定性图灵机上经过多项式时间得到验证。那么在非确定性图灵机上多项式时间可解是不是在确定性图灵机上也多项式可解呢，或者是不是 NP 完全的问题可以在多项式时间内可解呢，也即 NP 是不是等于 P 呢？这个问题目前还不知道答

案，它被认为是当前数学和计算机科学最重要的问题之一。P？= NP 问题在数学家斯梅尔（Steve Smale）的十八个未解数学问题表和克雷研究所的七大数学问题中都排第三。斯梅尔表中排第二的庞加莱猜想已被证明，排第一的是黎曼猜想。NP 完全问题是那些在多项式时间内可验证的问题。以 TSP 问题为例，一旦知道一个解，我们总可以在多项式时间内验证这个解对不对。如果 P 不等于 NP，那么很多衍生的关乎平常生活的话题就都有了理论依据：证明一个定理比验证这个证明要难，写书比读书要难，等等。

一个算法的空间效率就用它需要多少存储空间来衡量，多项式空间就是高效率的，简称 PSPACE。很容易证明 $NP \subseteq PSPACE$：因为多项式时间，无论确定还是非确定，都跳不出多项式空间，但这个关系是不是真子集关系，现在仍然未知。我们甚至不知道 $P \subset PSPACE$ 是否成立。如果这个成立，就是时间比空间更值钱的证据。

除了时间和空间的测度外，还有其他复杂性的测度，蔡汀的算法信息论的一个核心概念就是程序复杂性，也就是把计算机程序的长度当作复杂性的度量。

理论计算机科学家洪加威在 20 世纪 80 年代提出了相似性原则：计算装置之间互相模拟的成本是多项式的，也就是说靠谱的计算装置之间并不存在原则上的差异。相似性原则，类似于丘奇–图灵论题，是观察而不是数学定理。洪加威在他的原创论文和后来的专著中，证明了若干种重要的计算模型之间都是可以在多项式时间内互相模拟的。但我们不知道会不会有新的模型不服从这个原则。



洪加威（1937—　）

在证明了几种不同的并行计算模型互相模拟的性质后，洪加威指出并行计算模型也服从相似性原则。其实早期的计算理论文献中就有多头图灵机，多头图灵机可被单头多带图灵机模拟，而单头多带图灵机又可被单头单带图灵机模拟，这些模拟过程都可在多项式时间内完成。

洪加威 1960 年毕业于北大数学系，最后一年定专业为数理逻辑。他曾经想读吉林大学王湘浩的研究生，但未果，最后跟随段学复学了代数。1974 年北京市成立计算中心，他被调入。后来有人把他推荐给王浩，王浩又把他推荐给自己的学生库克，开启了洪加威的理论计算机科学之旅，时间不长但富有成果。他 20 世纪 80 年代中期曾回国短暂担任北京计算机学院院长，但又到美国周游列校，先后访问卡内基梅隆大学、斯坦福大学、麻省大学、康奈尔大学、加州大学伯克利分校和罗切斯特大学，每校一年，被访学校的理论计算机科学家对洪加威无不佩服。

相似性原则在刚提出时曾经轰动一时。和丘奇-图灵论题类似，相似性原则也不是数学定理，更像是物理定律。相似性原则后来被渐渐淡忘，乃至忽视，主要原因大概是这个原则已经变成计算理论的工作假设，大家已经习以为常。如果相似性原则不成立，那么大部分算法复杂性的成果都要被推翻了。另外，复杂性问题的哲学涵义远不如可计算性那么令人瞩目。

阿伦森（Scott Aaronson）认为复杂性实际上要比可计算性有更多的哲学涵义。

3. 超计算

丘奇–图灵论题的一个自然结果就是不可能存在比图灵机更强的计算装置。那么假设存在一种装置超越图灵机，会出现什么结果呢？图灵研究专家寇普兰（Jack Copeland）是逻辑学家，也是史学家，他把在可计算性上超越图灵机的装置叫“超计算”（hyper computation）。注意，这和“超级计算”（super-computing）不同，超级计算是量变，而超计算是质变。最近国内有所谓“超智能”（hyper-intelligence）的提法，但没有明确定义，不知和牛津未来学家博斯特罗姆（Nick Bolstrom）的“超级智能”（super-intelligence）有啥区别。有的人会用不同的术语指称“超计算”，例如彭罗斯就用“非算法”（non-algorithmic），还有人用“超图灵”（Trans-Turing或者 Super-Turing）。其实图灵本人就曾提出过“天启”（oracle）的思想：一个图灵机可以问 oracle 任何问题。当 oracle 的能力超过图灵机时，这个图灵机就有了超计算的能力。oracle 的存在有理论的方便性，但具备超计算能力的 oracle 本身还都不存在物理实现。

从图灵机看人工智能，要考虑两个层面，一个是可计算性，另一个是计算复杂性。丘奇–图灵论题在可计算性层面，洪加威的相似性原则在计算复杂性层面。在此我提出“超计算”的另一个标准：在复杂性层面不服从相似性原则。有一种可能性：可计算性上满足丘奇–图灵论题，但在复杂性层面不服从相似性原则的情况，也就是说用图灵机可以模拟，但不能在多项式时间内模拟，这种情况也算超计算。有人说如果我们遵从丘奇–

图灵论题和相似性原则，那么所谓超计算就是类似永动机一样的东西。

就本章的标题而言，“人是不是机器”有两种可能的答案：其一，人是不是可被图灵机模拟，也即可计算的；其二，人是不是可被图灵机有效地模拟，也即在多项式资源限定下模拟。

到底存在超计算吗？寇普兰系统性地从超计算的角度看人工智能问题。这个方法可为回答“人是不是机器”这一问题提供一些新的洞察。如果我们把超计算的探索从神学降级到世俗，倒是经常可以看见数学家和理论物理学家的身影。目前比较有意义的超计算模型有 BSS 实数模型和量子计算机。

4. BSS 实数模型

BSS 是三个人姓氏的首字母。B 是雷诺尔・布卢姆（Lenore Blum），她是少见的女性数学家和理论计算机科学家，她的丈夫曼纽尔・布卢姆（Manuel Blum）是委内瑞拉计算机科学家、图灵奖获得者、明斯基的学生，他们的儿子阿乌林・布卢姆（Avrim Blum）也是理论计算机科学家，一家三口都毕业于麻省理工学院，目前又都在卡内基梅隆大学任教。第一个 S 是沙布（Michael Shub），曾任职 IBM，是理论计算机科学家。第二个 S 是斯梅尔（Steve Smale），数学家，菲尔兹奖和沃尔夫奖获得者。斯梅尔是其中的长者，他老树开新花，传奇多变的学术生涯最新的兴趣是计算理论。BSS 模型的主要用途是为数值计算中的算法分析提供理论基础。BSS 模型的一个很大假设是，任意精度的实数四则运算可在单位时间内完成，这在数值分析中是有用而又方便的假设，但目前尚不知道如何在物理上实现这个。其实即使在数值分析之外，我们经常做类似的假设，例如，在排序算

法分析中，任意精度的数（可能是实数）之间的比较是单位时间的。

在 BSS 中，一阶逻辑的所有东西都是可判定的。这和图灵机是截然不同的，图灵机停机问题就是不可判定的。BSS 和图灵机的这个本质区别可溯源到 20 世纪 30 年代初期。那时哥德尔证明了整数的一阶逻辑是不可判定的。但几乎在同时，塔尔斯基证明了实数的一阶理论（几何和代数）则是可判定的。我们可以说图灵机和 BSS 分别是哥德尔定理和塔尔斯基定理的计算体现。

有些复杂性的性质，BSS 也和图灵机不同。比如线性规划在图灵机上被证明是多项式时间的，但在 BSS 上，复杂度是啥，目前不知道。如果在 BSS 上可以找到线性规划的多项式时间的话，在图灵机上就可以找到强多项式时间算法。这个问题被斯梅尔称为最重要的计算机科学的理论问题。

20 世纪 80 年代初就有人证明三层以上的神经网络可以逼近任意连续函数。80 年代末期，贾德（Steve Judd）在他的博士论文里证明了三层以上的神经网络学习问题在图灵机上是 NP 完全的。作者本人则证明了在 BSS 模型上，类似的神经网络学习问题等价于线性规划问题。目前各种神经网络学习算法都是工程，鲜有科学，神经网络算法多是些经验算法外加调参数，从业人员也多数没有计算理论的训练。伴随暴发户和显学的必然是浮躁之气。在各种学习算法里，很少看到目前关于什么算法适合什么问题的理论指导。

目前来看，BSS 模型没法物理实现，在某些情况下，可作为模拟计算看待。BSS 至少为简化带实数的算法分析提供了工具。

5. 量子计算

就像 BSS 是一位数学大师晚年的学术亮点，而量子计算则是一位大神级物理学家的最终关注。费曼在 20 世纪 80 年代初受奇才弗雷德金（Edward Fredkin）的启发，研究量子计算，随后于 1983 年至 1985 年间，在加州理工学院开了一门“计算机的潜力和限度”的课程。这门课费曼自己讲了“计算理论”“可逆计算”“量子计算机”和“计算的物理问题”等选题。他还请了生物物理学家霍普菲尔德（就是发明 Hopfield 网络而复兴神经网络的那位）和集成电路大佬米德（Carver Mead）等助阵。这门课被录了音，由听课的两位英国博士后整理成文字出版，书名《费曼计算机科学讲义》，可惜费曼没等到书面世就病逝了。这书包罗了费曼自己的讲课内容，而外援助阵的那部分一直没机会成集。

费曼并不是最早考虑量子计算的人。IBM 是计算物理学[①]的源头。计算的物理学研究有实际需求。从计算的角度看，图灵机只有数学约束而没有物理约束。一个可能的物理约束是能量：图灵机的读写头和纸带的运动是需要能量的。现代计算机的组件是逻辑门，有两种门，一种是可逆的，如“非门”，因为 A 的非的非还是 A；还有一种是不可逆的，如“与门”，因为从“A and B”的结果推不出 A 和 B 的原样。IBM 的物理学家朗道尔（Rolf Landauer）在 1961 年提出了朗道尔原理：任何不可逆计算都需要能量。同在 IBM 的另一位物理学家本内特（Charles Bennett）在 20 世纪 70 年代提出可逆运算不需要能量，并证明对任何图灵机都能找到一个对应的可逆版本，能实现同样功能而不损失效率。天才弗雷德金首先提出可实现

① 这里指的是用物理研究计算的 Physics of Computation，而不是用计算研究物理的 Computational Physics。

的可逆“弗雷德金门”，正是他引导费曼关注计算的物理学问题。这个只在加州理工学院上过一年大学的家伙后来成为成功的企业家，居然一度（1971 年至 1974 年）当过麻省理工学院 MAC 项目的头儿。MAC 的创始人是麦卡锡和明斯基，著名的 AI 实验室和计算机科学实验室都是从 MAC 分出来的，2003 年又合并为计算机科学和 AI 实验室（CSAIL）。

费曼考虑的问题是如何以任意精度来模拟一个物理系统。费曼的方法是构造一台量子计算机，它求解问题的时间不随问题的规模呈指数增长。量子计算并不是一步一步的经典计算，而只是测量系统的输出结果。

最早提出通用量子计算模型的是以色列出生的英国人多依奇，他 1985 年在《皇家学会会刊》发表文章首创通用量子图灵机模型，为量子计算提供了理论基础。他进一步提出丘奇–图灵–多依奇原理（Church-Turing-Deutsch Principle）：任何物理过程都可以被通用计算装置（量子计算机）所模拟。与丘奇–图灵论题不同的是，它在计算里添加了物理内容。多依奇认为经典图灵机的约束是经典物理，而量子图灵机的约束是量子物理。

多伊奇认为不可预测性（例如蝴蝶效应）和难解性是本质上完全不同的两种问题。不可预测性是由初始条件的微小变动引起的系统不稳定。而难解性是由内在的计算量导致的。费曼认为测量本身也是一种计算。当计算量很大时，最简单的方式是让自然界自己该干啥干啥，而在对的时刻测测结果就可以了。举一个不精当的比喻，想知道子弹的弹道，一种方式是考虑所有可能外部内部因素，依靠计算；另一种就是让子弹飞，然后测量。随机数可以通过伪随机函数生成，也可以通过测量一些噪声源得到。图灵 1949 年就研究过通过外部电子噪声源得到随机数的方法。

在图灵机上很难求解的问题有可能在量子计算机上用多项式时间解

决。其中最热门的问题是素数分解。素数分解问题是公钥加密算法 RSA 的基础。一般认为素数分解是很难的问题，也就是说当一个数很大时，把这个数分解为素数要消耗的计算资源很大。素数分解问题是不是 NP 完全的，目前还不知道，即使不是 NP 完全的，至少也是很难在多项式时间内完成求解的问题。要把一个数 n 分解为素数，最直观的算法是筛法，即将 n 被从 1 到 $\sqrt{n}$ 的数去除一下，RSA 算法目前密钥的典型长度是 1024 位二进制数，朴素筛法就要大约 2^{512} 次运算，事实上，目前最快的素数算法要远快过朴素筛法，但仍然是近指数的。

彼得·肖（Peter Shor）1994 年给出了在量子计算机上实现素数分解的有效算法。2001 年 IBM 在一台用核磁共振技术实现的 7 个 qubit 的量子计算机上把 15 分解为 3×5。中国科大在该领域也是领先者之一。

量子计算机不能求解图灵机不可计算的问题，从某种意义上说，量子计算机满足丘奇–图灵论题，但在某些条件下与相似性原则冲突。大规模的量子计算机的实现，目前仍存实现的问题。有人认为 1024 位密钥在不远的将来会被量子计算机破解。也有对量子计算不买账的，其中就有和库克独立发现 NP 完全性的列文（Leonid Levin）和理论计算机科学家古德瑞克（Oded Goldreich①）。他们认为不可能造出实用的、成规模的量子计算机。他们并没有被肖的素数分解算法打动。毕竟素数分解的复杂性还不明了。反而一些理论物理学家更加关注量子计算机的发展，他们其实并不关心能否造出一台可用的量子计算机，他们期盼这个过程中可能出现的理论问题。

① 他同两位图灵奖获得者 Silvio Micali 和 Shafi Goldwasser 紧密合作，但他自己并未获奖。

6. 计算理论的哲学寓意

多依奇的野心是把量子物理、计算理论（丘奇–图灵论题）、进化论和知识论（波普尔的“知识的增长”）融合到一起。人工智能的所有根本问题是他整个大融合理论的自然推论。他批评那些反人工智能的人，说他们的论点总是抓不到重点。他说只有彭罗斯抓到了点子：反对人工智能必须从推翻丘奇–图灵论题下手，尽管多依奇不认同彭罗斯的立场。

多依奇批判库恩的“范式”，他观察到即使新学已经成为主流理论并替代旧学，但新的范式并没有彰显，人们仍然并没有新学的基本原理当作工作假设。他以人工智能为例，那些反对人工智能的人并没有意识到他们在和自己学科的基本原理作对。我觉得多依奇操之过急了。大多数现在人云亦云搞深度学习的人，压根就没听说过丘奇–图灵论题，更谈不到理解并应用甚至下意识地反对这个论题了。科学发展太快，即使在一个大学科内，大圈子内的人也未必能够了解所有小圈子的活动。

丘奇–图灵论题加上相似性原则可以解释很多心灵哲学的学派。马尔（David Marr）的视觉理论被认为是计算主义的代表。马尔有个三层假设，首先是计算层面，其次是算法或表示层面，最后是物理实现层面。计算层面对应的是丘奇–图灵论题，算法层面对应的是相似性原则。

丘奇–图灵论题也有功能主义的解释：丘奇的 λ 演算更象是高级的程序语言，而图灵机则像是低级的物理实现，但它们确实是可以互相模拟的。基于 λ 演算的变成语言例如 LISP 就是符号处理，但图灵机则更像是汇编语言，它的符号集只有 0 和 1。

当人工智能的成就被当作“算法的胜利”和“自由意志的终结”时，我们要小心，被终结的是图灵机的自由意志，还是量子计算机的自由意志，

抑或是 BSS 的自由意志，是可计算性的自由意志还是复杂性的自由意志。我们不能指望畅销书作家代替我们回答，而是要自己思考自己回答。

计算理论家阿伦森曾写过一篇“为什么哲学家应该学点计算复杂性？”，我们也可以换个标题“为什么人工智能学者应该学点计算理论？”人们常说是明斯基和佩珀特的《感知机》（*Perceptrons*）一书导致了神经网络研究近 20 年的衰败，但神经网络的研究者不该反省下自己数学和计算理论功底的缺乏吗？从当下人工智能的浮夸风气中，看不到教训被吸取。

7. 丘奇–图灵论题、超计算和人工智能

如果我们认可丘奇–图灵论题和相似性原则，那么人就是图灵机。所有目前的人工智能工作都是建立在这个认同之上的。丘奇–图灵论题和洪加威相似性原则是战斗在第一线的研究者的工作假设。丘奇–图灵论题说的是能行性（effectiveness），而相似性原则则是说的效率（efficiency）。

多依奇在他的科普著作里提到，他认为人类有史以来有 4 个伟大理论：达尔文进化论、波普尔证伪理论、量子理论和计算理论。他的工作把量子理论和计算理论整合到一起。但他不认为人脑是量子计算机，这点有别于彭罗斯。

自然科学和唯物论、经验论的关系要比唯心论、理性论的关系更近，背后都有一条归约主义路线图（reductionist），从生物到化学，再到物理。一个有机体最终被归约到物理化学过程，一个活的、有意识的生物最终会被归约到神经网络过程。这是自顶向下的思路。

近来也有自底向上的思路，例如细胞自动机，利用很简单的几条规则，就可展示很复杂的行为。沃尔夫勒姆在他的《新科学》（*A New Kind of*

Science）一书中提到的这种现象其实早就被数学家康韦（John Conway）观察到，他设计了《生命游戏》（*Game of Life*），企图利用细胞自动机来说明确定性和自由意志的问题。高德纳在评论康韦的工作时说：所有规则都是确定性的，但游戏的演进过程却给人一种自主性的感觉。高老喜欢阅读英国女作家塞耶斯（Dorothy Sayers），她更偏爱写剧本而不是小说，她说剧本给了演员发挥再创作的机会，这个再创作就是自由意志。高老说量子力学为自由意志提供了空间，也使得上帝可以操纵世界而不违反物理定律。

所谓“太阳底下没啥是新鲜的”，但几条简单规则展示的行为却无法解释。如果考虑数学定理证明，我们可以说勾股定理不新鲜——毕竟从简单几何公理不用费太大力气就可证明。但我们敢说黎曼猜想也不新鲜或者庞加莱猜想的证明也不新鲜因为所有结果不都是可以从起始点（那几条数论公理）推出吗？这似乎模糊了柏拉图主义（实在论）和构造主义的边界。

机器人学者布鲁克斯（Rodney Brooks）曾在《自然》杂志上撰文讨论生命体和物质的关系，他认为，对于生物体（living systems），我们也许并不缺必要的物理学，但缺新数学（unimaginable “new mathematics”）。每次有新的数学工具被发明时，例如混沌、小波等，都会被先用来解释生物系统。生命体的物理性质通过计算手段来模拟是很贵的。也许有一种我们目前还不知道的新数学，能最终解释生命体的各种现象，这种数学可能是和计算相关的。布鲁克斯没有说这种新数学是不是超计算的。

如果超计算是可以实现的，我们仍然不能确定人不是机器，但我们至少有了人要比经典图灵机高明的可能性证据。彭罗斯就是企图用量子计算来否定基于经典图灵机的人工智能。彭罗斯除了不满意强人工智能之

外，他似乎也不满某些物理学家，我猜可能是费曼和弗雷德金。他在《皇帝的新脑》中说，哲学家被计算机科学家误导，而计算机科学家又被物理学家误导，但物理学中则开始流行“所有东西都是计算机”的有害思想，这是不指名地揶揄弗雷德金和深受其影响的费曼。无论如何，我对费曼和彭罗斯都存有智力的敬仰，他们随便学学就能得知计算机科学的精髓（看看《费曼计算机科学讲义》和《皇帝新脑》中对理论计算机科学的论述），而我很少看到计算机科学家敢对理论物理说三道四，姚期智大概是唯一的例外。

纯逻辑的问题，图灵机和人没啥大区别，无论可计算性还是复杂性。但对非逻辑问题，彭罗斯想说人可能相当于量子计算，那就可能不等价于图灵机了。但深究起来，这里面还有两个跳跃（gap），第一图灵机到量子计算的跳跃，第二，量子计算到人的跳跃。

目前叫嚣声浪很高的“超级智能”（super-intelligence）是个模糊不清的伪概念。如果我们认可图灵机等价于智能，那么超计算或超图灵自然可以超智能；即使没有超计算，依靠硬件的摩尔定律和算法的改进，仍然可以达到超智能。如果不认可图灵机等价于智能，那么智能本身就是一种超计算，超智能自然就能超计算。一种可能性是更厉害的模型：超量子计算，在可计算性或复杂性上超越量子计算，另一种可能性是超智能只是更快的量子计算。按照彭罗斯的说法，智能就是量子计算，甚至超量子计算。

弗雷德金说，他既不是有神论者也不是无神论者也不是不可知论者。如果从他的“数字物理”（Digital Physics）、“数字力学”（Digital Mechanics）、“计算宇宙”（computing universe）的观点看，倒可以把他归为“计算论者”（computationalist），尽管这个词主要是用在认识论而不是宗教。

在图灵曾任职的布莱彻利庄园里，computer 一词就是指那些人类计算

者（通常是女性），她们按照算法设计师的要求，执行计算步骤来完成一个计算任务。也许人类和计算机并没有什么本质差别，只是给 computer 一词的赋值不同罢了。

参考文献指南

计算理论需要很多数学背景，想进一步了解可参阅 Arora (2009) 或 Sipser (2012)，但必须警告在先：水很深，非数学或计算机专业的，非诚勿扰。Hong Jia-wei (1986) 是洪加威本人关于相似性原则的原始文献。Blum, Cucker, Shub and Smale (1998)是实数理论原创者的权威论述。Feynman (2000) 和 Milburn (1999) 是量子计算的导论，可读性介于科普和专业之间，而 Deutsch (1998)和 Penrose (1989) 则更偏科普。

对于相关的哲学和历史感兴趣的读者，关于图灵的几本文集是必看的：Copeland (2013)、Cooper and Leeuwen (2013)、Cooper and Hodges (2016)、Herken (1994)、Petzold (2008)。想对人工智能说三道四的哲学家和媒体人，至少应该看看计算理论的新星阿伦森的几篇文章，相比费曼和惠勒那些对反智深恶痛绝的物理学家，阿伦森显得更有耐心也更宽容。

第 11 章 智能的进化

Science is what we understand well enough to explain to a computer.
Art is everything else we do.
科学就是那些我们能对计算机说明白的东西，余下的都叫艺术。
——Donald Knuth（高德纳）

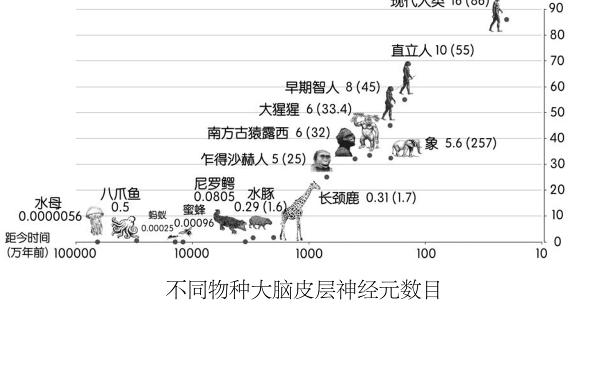
1. 大脑的进化

人脑并不是所有动物中脑子最大的，大象和鲸鱼的脑子都比人脑大。那么人为什么比其他动物更聪明呢？神经生物学家会告诉你，虽然人脑并不是最大的，但是是最好的，好坏的衡量标准是脑中神经元的数目。但人脑有多少神经元呢？我们经常在教科书或科普杂志上读到的数字是 1000 亿。这个说法尽管广泛流传，但其实是以讹传讹的结果。某一天巴西籍神经生物学家埃尔库拉诺–乌泽尔（Suzana Herculano-Houzel）问了自己这个问题后，遍查科技文献，发现并没有现成的科学标准答案。于是她成了第一个研究这个重要问题的科学家。某些重要的科学问题貌似平常，科学家都想当然一定有其他人早就研究过这个问题。埃尔库拉诺–乌泽尔花了几年时间找到了答案，她在 2005 年至 2015 年陆续发表了几篇文章，奠定了她在学术界的地位。2013 年她成为第一位巴西籍的 TED 演讲人，她的声名也从此为大众所知。她 2016 年离开巴西的里约热内卢联邦大学前往美

国范德比尔特大学任教，同年她的研究成果出版成书 *Human Advantage: How Our Brains Became Remarkable*，该书很快成为畅销书，并被翻译为多种语言。2017 年该书中文版以《最强大脑》为题出版。她创造的“大脑汤”（brain soup）的方法最终使她成功地测定不同动物大脑的神经元数量。书中不仅有研究成果，还有更有意思的研究过程，包括她是如何把大象的大脑从非洲弄到美洲的新奇故事。当然，这本书的两个重要科学结论更值得讨论。

首先，不同动物的脑子构造也有所不同，脑中的神经元数量也完全不同，人脑中总共有 860 亿个神经元，其中大脑皮层有 160 亿个神经元。大脑皮层的神经元数量决定了动物的智力水平，人的大脑皮层中神经元数量远高于其他物种，所以人类比其他物种更聪明。大象的脑子总共有 2570 亿个神经元，但是其中 98%的神经元都存在于大象的小脑中。而大象的大脑皮层只有 56 亿个神经元，无法与人类相比。

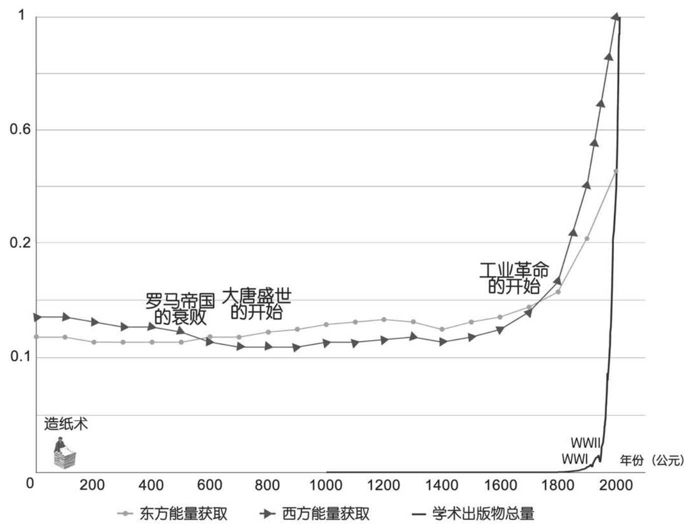
其次，大脑皮层中的神经元数量越大，能耗也越大。人脑每天消耗的能量占人体全部耗能的 25%。人之所以能够很快超越其他物种，主要是因为人类掌握了烹饪技术。能够在短时间内摄入大量卡路里以支持大脑运转。其他物种则将摄入的卡路里用于维持身体运转，不得不牺牲大脑皮层的神经元数量。直立行走的动物要比四足动物更省能耗。



不同物种大脑皮层神经元数目

2. 能源的摄取和消耗

说到能耗，我们不得不说说近几年出的另一本有意思的书《西方将主宰多久》(*Why the West Rules—For Now*)。这书的作者是美国斯坦福大学的历史学教授莫里斯 (Ian Morris)。他用能耗作为主要测量参数比较了东西方文明，他的起点是 15000 年前。他的结论是西方除了在中世纪 (公元 500 年到公元 1500 年) 的大约 1000 年期间外，一直都是领先东方的。



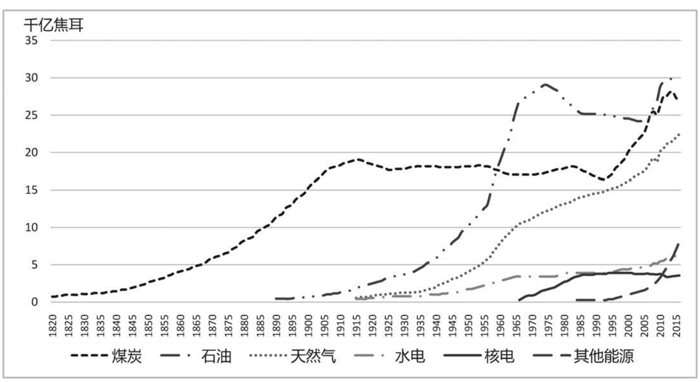
东西方能量摄取和学术出版物数量趋势对比

这个结论从一个方面间接回答了李约瑟问题[①]。我们从来不会问为什么非洲没有出现科学，李约瑟问题给人一种误解和错觉：东方曾经发达过，因而有潜在诞生科学的土壤。事实上，东方真正发达起来是在 19 世纪中

① 李约瑟问题："尽管中国古代对人类科技发展做出了很多重要贡献，但为什么科学和工业革命没有在近代的中国发生？"

叶之后，西方教会了东方新的能量摄取手段。这让人不禁想起日本的黑船事件和中国的鸦片战争。这书的结论也不是没有争议，加拿大新不伦瑞克大学的历史系教授杜彻斯内（Ricardo Duchesne）就撰文批评莫里斯的东西方定义不合常规。莫里斯把东方界定为东亚和南亚，于是东方包含中日韩，这还算能接受；而莫里斯的西方定义却是欧美加上伊斯兰。莫里斯坚持自己的观点，而他的这种观点并不是他的原创。事实上早在 1987 年，美国学者威尔金森（David Wilkinson）就提出类似的西方文明圈的界定方法。哈佛的政治学家亨亭顿（Samuel Huntington）在《文明的冲突》（*Clash of Civilizations*）中的三分法（东方、西方、伊斯兰）是更为常规的说法。我们不禁好奇，如果采用三分法，莫里斯会得出什么结论。不知这类研究会不会也受到政治正确性的影响。不管几分法，印度都是一个异数，既不算西方也不算东方。考古学家兼东方学者斯坦因（Aurel Stein）就曾指出，印度不能被简单地以东西方来划分。

随着科技的进步，人类摄取能源的成本逐步降低，能源的形态也呈多样性，从最早的碳能源逐步过渡到清洁能源。但在工业革命后，总的能源摄取和消耗是呈指数上升的。

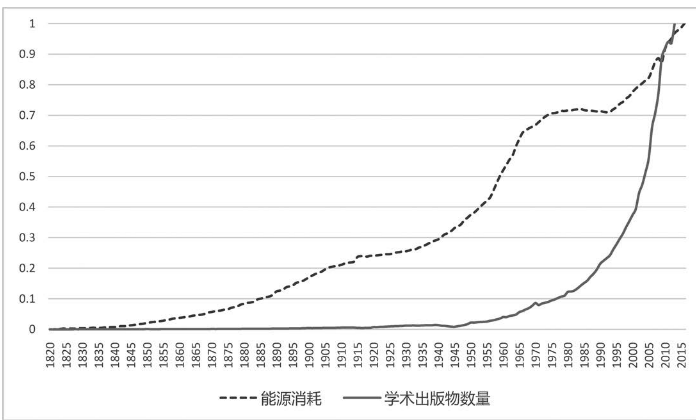


不同能源的分布

3. 全社会的算力作为文明的测度

埃尔库拉诺–乌泽尔的研究给出了人类智力高于其他物种的有力证据。智人的出现大约在十万年前。莫里斯用能耗来测度不同文明进化的过程，他的时间尺度是以万年计。如果我们把人类文明的尺度进一步缩小，只看过去千年或百年的比较，信息处理能力可以成为测度指标。具体地说，信息处理能力可以用计算或存储能力衡量。

如果我们宽泛地定义信息处理能力，印刷和出版是最早的信息处理和存储手段。关于印刷术的原创到底始于西方还是东方，没必要争论，但印刷术对西方文明的加速要超出东方文明应是不争的事实。中国的现代印刷术是 19 世纪末期从东南亚引进的，当时的主因是印刷报纸。如果我们对比能耗和出版物数量，可以清楚出版物的增速要远超能耗的增速。

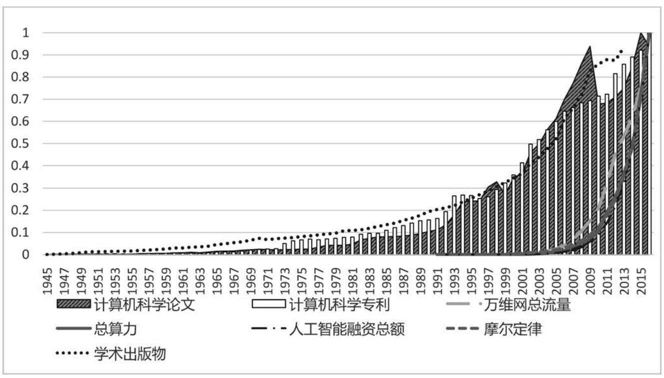


总能耗与学术出版物

当我们把时间尺度更进一步缩小到几十年的范围，我们可以用社会的计算能力来测度文明。从算筹到中式算盘，漫长的数千年中计算速度的

增长并不明显，一次加法的速度从十秒级到了秒级。机械计算机提升了计算速度。而电子计算机诞生之后的计算速度的增加可以用摩尔定律来粗略衡量。摩尔定律说的是每 18 个月，计算机的速度加倍，存储能力加倍。这是一个指数型增长，其发展速度超过了人类历史上任何一项技术。

随着信息技术对人类生活的进一步渗透，全社会每人拥有的计算设备的数量也呈指数增加。想想我们最早只有个人电脑，然后又有各种可穿戴设备和嵌入式设备。每个现在的可穿戴设备的计算能力都超过了上一个时代的超级计算机。而全社会的算力应该是全社会计算设备的数目与单台设备计算能力的乘积。这是两个指数曲线的叠加。我们由此有了对大规模数据的处理能力，加速了社会信息化的进程。



专利、论文、万维网流量、社会算力、人工智能全球融资总额、摩尔定律及出版物的增长趋势对比

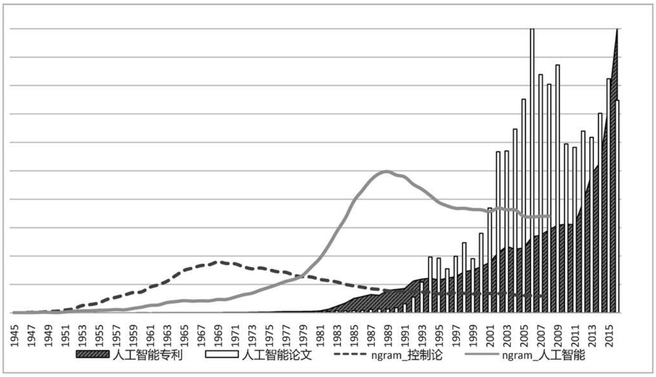
与计算机能力同时提升的是数据量。最近十几年时间里产生的数据量，足以使之前人类历史积累的数据量忽略不计。信息爆炸的数据源从微观世界到宏观宇宙，无所不包。从分子生物学到航天技术和天体物理，数据的累计代表了人类对外部世界更深刻的了解。工业革命加剧了知识的累计。

互联网和社交媒体产生的数据更是“秒新时异”。互联网的总流量在开始的几年远超摩尔定律的增速，带动了信息存储和社会算力的提升。

4. 人工智能从哪里来?

当下在第一线做研究的人，能接触到的历史要么太大，要么太小。一个科学家关心的时间颗粒度大概在五年左右，有时把时间颗粒度拉大，会让人更清楚地看到学科的发展脉路，有助于对学科进行宏观把握。大部分教科书顶多也就是把时间定在十年左右，太早的要么丢掉，要么不去梳理来龙去脉。我们看到的“大历史”，时间颗粒度太大，基本单位变成了范式，历史过程成了范式迁移，或者阶级斗争。“范式”这种概念对历史学家是必需的，但对当下的内行研究者没啥用。比“范式”小一号的概念是哲学家波普尔的“猜想与反驳”或“批判”；而比波普尔更小一号的是被波普尔批判的归纳法。当然，从机器学习的角度可以再进一步外推，比归纳法更小一号的是深度学习。

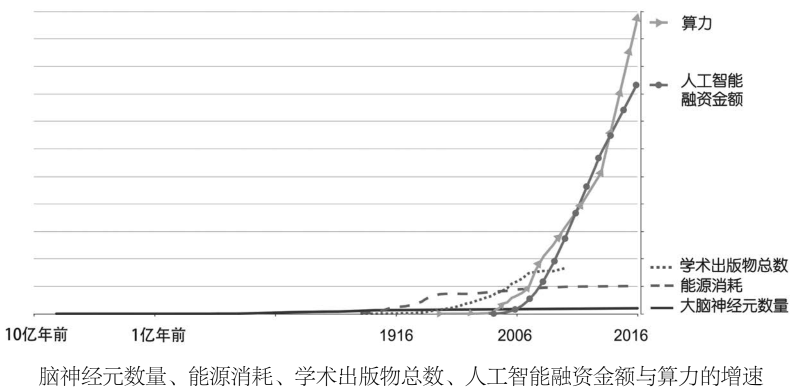
如果以这样的时间颗粒度作为分类标准的话，人工智能大概可以算是范式级别的。人工智能如果从控制论算起，用谷歌 Ngram 衡量，那么目前经历了两次高潮，控制论之后的第二次浪潮是在 20 世纪 80 年代，那时正是专家系统和日本第五代计算机项目得势的时候。但进入 90 年代后，人工智能又呈现出下滑趋势。现在的新一轮高潮是进入 21 世纪之后开始的，2006 年辛顿在《科学》上发表的深度学习相关文章，不知算是起因还是巧合。如果按照麦卡锡的说法，控制论不算人工智能的话，现在（2017 年）尚处人工智能的第二次高潮，我们不知道这一次还能持续多长时间。



人工智能专利、论文及 Ngram 人工智能曲线

5. 人工智能向哪里去：会有超级智能吗？

如果我们把大脑神经元数量、能源消耗、学术出版物总数、人工智能融资金额与算力的增速做个对比，可以看到一个有趣的现象，能源消耗的增长速度比大脑神经元数量的增长速度更快，而学术出版物总数的增长速度又比能源消耗的增长速度更快；人工智能融资规模与算力的增长速度相当，均高于其他几个维度的增速。



脑神经元数量、能源消耗、学术出版物总数、人工智能融资金额与算力的增速

2016 年 AlphaGo 对李世石的胜利引起了新一轮对超级智能的讨论，焦点是将来会不会有一个全新的物种在智能上全面超越人类。牛津的哲学家博斯特罗姆（Nick Bolstrom）在 2014 年写了本未来学的著作《超级智能》，讲到了人类在面临不断发展的机器时代的存在危机。物理学家霍金和企业家马斯克都附和博斯特罗姆的立场。

畅销书《未来简史》（*Homo Deus: A Brief History of Tomorrow*）的作者、另类历史学家赫拉利（Yuval Harari）为这个潜在的新物种起了个有意思的新名，叫"神人"（Homo Deus），恰是这本书的英文书名。"神人"就是超级智能的主体。赫拉利真像是人工智能界安插在人文学科里的"托儿"，在潜伏期到了之后，出现得真是时候。历史学家本来应是总结过去的，但赫拉利却偏要预测未来。赫拉利的前一本书《人类简史》从七万年前的认知革命一直讲到当下，为《未来简史》做了铺垫。用不同的时间颗粒度看待过去，会得到不同的结论。《尤利西斯》中的几个小时，茨威格作品中人物的一生，或赫拉利的七万年，关心不同的过程。颗粒度也可以是主体的，一个基因，一个人，一个群体，不一定非得是一个小的物质颗粒只配得上小的时间单位。想想基因人类学，基因在几万年的空间分布，帮我们了解人类的起源和迁移。当用太大的颗粒度研究历史时，历史学家的用处会令人质疑，那本该是初中历史课本的使命，任何结论都不会令人惊奇。

在人工智能研究的早期，就有 AI 和 IA（智能辅助 Intelligent Assistant，或智能增强 Intelligence Augmentation）之分。而人工智能又有强弱之别，强人工智能是说机器会全面达到人的智能，强人工智能有时也会和"通用人工智能"（Artificial General Intelligence）同义；而弱人工智能是说机器会在某些方面达到人的智能——AlphaGo 就是弱人工智能的代表。目前不会再有人质疑弱人工智能了，但强人工智能立场的主张者却面临各种挑战，

其中之一是不同智能功能的整合。机器人可以比人更快地跑步，更好地下棋，更准确地识别语音和图片。如果功能整合是可能的，那我们可以想象，比如谷歌的 AlphaGo 整合了 IBM 的沃森（Watson），除了下棋赢人类，还能通过美国行医执照考试。更进一步，如果把这个混合物架在波士顿动力公司（Boston Dynamics）的机器狗上，它还能 5 秒就跑个百米。到底是感知更难还是认知更难，这又是一个长话题。曾经有人说，人对机器的最简单控制就是断电，但现在最简单的扫地机器人也知道快没电时找回基座充电。把人工智能教科书中的 *n* 种智能功能整合起来（例如下棋、图像识别、语音识别、规划等），是不是就会达到强人工智能或者超智能？

目前人工智能系统功能的单一性会不会只是一种错觉？乐观主义者会认为弱 AI 向强 AI 的过程是个进步的过程，虽不是一夜之间。《终极算法》（*Master Algorithm*）的作者代表了一类乐观主义者，他认为会有一种统一的终极的机器学习算法，只要机器按照这个算法一直学下去，某一天就会超过人类。他给出的五类算法是进化的、连接主义的、符号的、贝叶斯的和类比的。作者的研究领域是如何把这些算法统一起来。我们不禁怀疑，即使把这些算法都能整合起来，是不是就构成了终极算法呢？难道不会有新的算法被发明出来吗？例如作者列的单子里明显缺失了强化学习，这是一个很老的算法，在 2016 年随着 AlphaGo 才得以扬名。有些机器学习的从业者缺乏对计算理论的了解。他们不知道面对这样的问题应该怎么办。“终极算法”的提倡者应该借鉴图灵机和丘奇–图灵论题这样令人信服的理论方法。悲观主义者的立场正在退缩：在每一个单独的领域，都有可能超越人类，不一定会在整体全面超越人类。如果这个整合的整体在我们可以想到的各个方面都超过人类，那我们如何对付这个新时代的“弗兰肯斯坦”。

人工智能作为学科，经历了几次大起大落。每一次的高潮都是一个旧哲学思想的技术再包装，而每一次的衰败都源自高潮时期的承诺不能兑现。这一次的浮夸轮到了机器学习，被压制多年的机器学习研究者们号称，尽管机器学习是人工智能的子学科，但很快就会独立，并且会比人工智能更宏伟。他们说人工智能是让机器达到人的水平，而机器学习是让机器超过人的水平。这倒是一个省事的赶超。还有人宣称甚至科学家的职业也将会消失——因为机器可以学得更快，但计算机科学家例外。我倒是怀疑计算机科学家是不是现在就应该组织工会。日本第五代计算机的失败似乎从来没有发生过。历史学家都忙着参与预测未来，当下是一个不需要总结教训的时代。AlphaGo 在输给李世石的那个晚上，可以在自己和自己不断对弈的过程中学习并克服自己的弱点。无论对错，机器学习的速度确实有可能快过人类进化的速度。

过去的机器旨在节省人的体力，现在的机器开始代替人的智力。人通过两性繁殖的进化速度远远赶不上机器。机器的进化速度服从摩尔定律——每 18 个月性能提升一倍，而人的进化速度则是 20 年一代人。人作为物种，是不是不再具备进化的竞争优势？依靠硬件的摩尔定律，是不是可以达到超级智能？

新的智能存在可以是人工智能的“主体”（agent），也可以是生物学意义上的物种。2016 年夏，在美国公共电台（Public Radio）热议的一个话题是，通过修复一个受精卵的一小段染色体，就可以避免或治疗某种疾病。这是一个真实的伦理问题，因为已经有这样的病例发生。如果孩子出生，那么他/她的父母是谁？反对方的理由也很简单，多小算是“一小段”，1% 还是 49%？甚至更进一步：可不可以有更多不同来源的基因参与？英国《经济学人》2017 年 2 月的一期封面标题就是“Sex and Science”，基因编辑

是热门话题。这种“转基因”疗法如果目的不是为了治病，而是作为物种改良的手段，人类可以接受吗？

图灵在 1936 年那篇文章“论可计算的数”中证明了图灵机和其他计算装置的等价性，并由此有了所谓丘奇–图灵论题，也就是说任何计算装置都等价于图灵机。这个论题不是数学定理，但却是整个计算机科学的基础。这个论题隐含着强人工智能的可能性：智能等价于图灵机。近年，也有人探索超计算（hyper-computation）：计算能力超越图灵机的装置。量子计算作为超计算的一种潜在设备，被彭罗斯用来论证人的智能要超越图灵机。

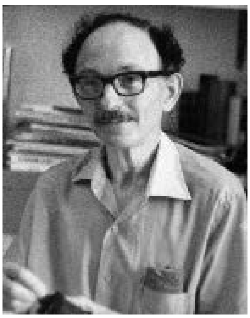
不懂计算理论的历史学家如赫拉利以及未来学家如博斯特罗姆，分不清“超计算”（hyper-computing）和“超级计算”（super-computing），自然也分不清“超智能”和“超级智能”（super-intelligence）。其实，他们所谓的“超级智能”早就被数学家和密码学家古德（Irving John Good）在 20 世纪 60 年代想过了，只不过那时“超级智能”被古德称为“智能爆炸”或 Ultra-intelligence，或“超人智能”。古德是图灵在布莱彻里庄园和后来曼彻斯特大学时的小伙伴。他 1967 年 7 月 7 日从英国搬到美国，他戏言他是“第七个年代，第七年，第七月，第七日，第七时到达黑堡的第七街区的，纯属偶然”。[①②] 古德后来加入了美国电影艺术与科学学会，因为他为大导演库布里克（Stanley Kubrick）1968 年的经典科幻片《2001 太空漫游》（*2001: A Space Odyssey*）担任过科技顾问。而他本人的原型也出现在 2014 年

① “I arrived in Blacksburg in the seventh hour of the seventh day of the seventh month of the year seven in the seventh decade, and I was put in Apartment 7 of Block 7...all by chance.”

② 我实在搞不明白为什么英文“1960s”要被规定翻译为“20 世纪 60 年代”，而不是更为简单明了的“1960 年代”。如果非要较真，那也应该翻译为“20 世纪第七年代”，哪来的“60”，一个“年代”可是“十年”啊，一个很简单的小学算术问题。

的以图灵为主角的历史题材电影《模仿游戏》（*Imitation Game*）里。

维纳曾经说："我们最好能够确认我们给机器设定的目的确实是我们想要的目的。"[①] 物理学家改行的科幻作家阿西莫夫曾提出机器人三定律，第一条就是机器不能伤害人，但"什么是伤害"本身就不好定义。AlphaGo 战胜李世石和柯洁，算是对他们的伤害吗？即使是科学家兼幻想家，也未必就能把不远的未来想明白。维纳和阿西莫夫可能都不懂计算理论。



古德（1916—2009）

图灵在 1950 年那篇被广为引用的文章"计算机与智能"的结尾处说：我们只能看到当下，但看见的这些就够我们忙活的了。[②]这真是实干型科学家和揣度型历史学家（speculative historian）的区别。

参考文献指南

本章的参考文献在文中都已列出。需要指出的是，《西方将主宰多久》有个单独出版的阅读伴侣，书名叫《文明的度量》（*The Measure of Civilization*），

① "We had better be quite sure that the purpose put into the machine is the purpose which we really desire."

② "We can only see a short distance ahead, but we can see plenty there that needs to be done." 本书附录 4 另译为"初见前路近可至，细思百事说待忙。"

该书为原书提供了详实的数据依据和方法论，电子版的书名是 *Social Development*，可在该书作者莫里斯的网站上免费获得：http://ianmorris.org/docs/social-development.pdf。

几本书的中译本的书名非常令人困惑，叫《未来简史》和《最强大脑》的就有好几本。有几篇书评的作者把几本书的内容都写串了，真不知道他们看没看过原书。实在不解为什么博斯特罗姆和赫拉利的书会成为畅销书。这不是他们的原创观点，他们从中衍生出来的东西既无深度也无趣。

相较任何一门其他的学科，关于计算的历史的教科书式的书籍，数目令人吃惊得少。法国中学数学教师 Georges Ifrah 写过一本关于数和计算的书，先是被翻译为英文在英国出版，后来在美国出版时，被分成两本不同的书：一本是关于数的，*From One to Zero: A Universal History of Numbers*；另一本是关于计算的，*The Universal History of Computing: From the Abacus to the Quantum Computer*。尽管这两本书在科普杂志《科学美国人》受到好评，但《美国数学会公告》2002 年发表过科学史家道本（Joseph Dauben）的书评，对于这两本书，尤其是后一本，他提出了尖锐的批评。除了指出事实性错误外，道本还提到，写计算的历史，多少要知道一些关于这门学科的知识。如果我退休了，倒是可以再试试写一本《计算简史》。

第 12 章
当我们谈论生死时，我们在谈论什么？

I don’t want to achieve immortality through my work; I want to achieve immortality through not dying.

我才不想靠我的作品达到不朽，肉身不死才算不朽。

——Woody Allen（伍迪·艾伦）

Immortality.
I make my journey through eternity.
I keep the memory of you and me, inside.

不朽兮，
我向来世奔袭，
你我的记忆，在心中安息。

——Bee Gees（BG 兄弟乐队）

苏格拉底说：哲学家只研究“正在死”（dying）和“刚刚死”（being dead）。除了这个啥都不管。有一种说法是哲学起始于苏格拉底之死。苏格拉底因为三项罪名被判死刑：腐蚀雅典青年，不敬城邦和引入自己的新神。苏格拉底受审前一天恰好赶上雅典的“花船节”，祭祀的船要离开雅典再返航。花期，城邦要保持清洁，因而不能执行死刑，于是苏格拉底临死前有一段时间可以和学生们聊哲学。柏拉图据此写了四篇对话。耶稣之死和苏格拉底之死不同，耶稣完成了使命，苏格拉底留下了一堆问题。



苏格拉底之死

苏格拉底临死那一天和学生们的对话更像是群口相声。苏老开的毕竟不是德云社，弟子们来去自由。他早晨起来就开聊，死期是日落，话题就是生死，反正这一天不聊这个也得被这个烦。他说人追求真理的最大束缚就是肉体，为了得到终极智慧，灵魂必须超越肉体，也就是摆脱感官的限制。换句话说就是人必有一死。不知道这是指导学生，还是宽慰自己。苏格拉底自己也很矛盾，灵魂到底是不是存在，灵魂是永生还是彻底消失。他最后一天的谈话被当时的在场者斐多记录，后来再转述给别人，最终变成了柏拉图的《斐多篇》。这个一辈子教导别人的老师，最后这一篇算是自己对自己说的。

在苏格拉底的理论里，生死是一对范畴，死由生来，生也必缘于死，生死轮回。面临生死的态度，其实有神论和无神论并没什么区别，有区别的只是乐观派和悲观派。苏格拉底被宣判死刑后的申辩书被柏拉图写成《申辩篇》，最后一句是："该说再见了，我去死，你们接着活。生死究竟哪个更好，只有天知道。"

下面是一段假想的对话，发生在我和苏格拉底之间。尼克（Nick）本是希腊名字 Nickolas 的昵称，两个人对话，尽管时间错位，但毕竟都是希腊人，也不算太出格。蒯因的女儿在纪念蒯因逝世一周年的讲话里套用蒯因笔记里的一句话“未完待续”。这也是一段可以永远进行的对话。

苏：那么活着的意义是什么呢？

尼：我六岁时就在思考人生的意义，可能我是个悲观主义者，我那时得到的结论是：人活着就是为了不痛苦地死。我看到那些意志最坚定的人在面临死亡时，也无可奈何，甚至有些无神论者也在考虑皈依。冯诺伊曼临死时就皈依天主教了。

苏：那是因为他们没有找到安慰自己的方式，他们自己也不相信他们平日说教别人的陈词滥调。我有一套方法，你跟着我的思路来，也许你会变成一个乐观派的实干家。

尼：我太想听了。

苏：如果你断了一条胳膊，你还是你吗？

尼：当然是。

苏：如果断肢再植，再给你换一条别人的胳膊，你还是你吗？

尼：当然是。

苏：如果给你装一个功能相仿的假腿，你认为是你自己在行走呢？还是假腿在行走呢？

尼：我当然认为是我在行走。我知道你给我挖坑呢，就像有人会问“潜艇会游泳吗”。

苏：你上路了，再给你来一个：如果给你换一个别人的心脏呢？你还是你吧？

尼：当然是。不过，我知道你正给我下套呢。希钦斯曾说：“我不是有个身体。我就是个身体。”（I don't have a body. I am a body.）你开始肢解我的身体了。

苏：假设你的身体出了问题，但你的头颅还完好。并且有医生有本事把你的头换到别人的身体上，如果手术成功，你还是你吗？

尼：这个嘛。目前的理解是我之所以是我，是因为我的 DNA。照你这说法，DNA 可能是别人的了，但意识还是我的。功能和实现可以分开，意识是功能，身体只是实现。所以，如果我们认为我之所以是我是因为我的自我意识，那么我承认即使换了整个身体，我还是我。

苏：其实，从你 6 岁起到现在，你身体里的细胞都不知换了多少遍了。是什么东西让你从小到现在还保持连续性？

尼：是自我意识？自我意识就是记忆吧？我怎么觉得您现在更像是您的学生柏拉图，您更相信意识是不取决于各种感官的。

苏：那我们再前进一步：假设你的脑子也受到伤害，但你大脑里负责记忆的部分还在。对了，我离开你们太久了，那部分叫啥？

尼：海马体和前额叶。

苏：假设你的海马体和前额叶还完好，那么把你的记忆体移植到别人的大脑里，你还是你吗？你前一天躺在手术室里，第二天，你醒过来，你发现你的面貌和身体全换了，你觉得你还是你吗？

尼：这个嘛，你在试我的底线呢。让我想想……你曾经说过：“所有的学习，都是记忆。”所有的学习，甭管多深，都是记忆。

苏：在你昏睡过去和醒来之前，你算死了吗？

尼：真不好说。阿西莫夫说死后的人生就是无梦的睡眠。这只是个比喻，并没有实质的内容。

苏：这一睡一醒，算是濒死经验吗？

尼：我猜，我醒来还是我，或者说，那个我醒来所在的身体上带着我的意识，如果是这样的话，我没死。

苏：你没死和你是不是原来的你是同一个问题吗？

尼：我醒来也许神经突触都联在一台超级计算机上，就像普特南的“缸中脑”。如果有了灵魂寄生的技术，我还是不是原来的我，这个问题不重要了。

苏：如果你醒来的那个身体是个异性呢？

尼：我猜我还是我，尽管我的生活会因此不便，我就只当我同时做了整容和变性手术，但我还记得我的父母，我的孩子，我的恋人，我的朋友，甚至个把敌人。当你保留美好甜蜜温柔的记忆的同时，那些残忍痛苦也夹杂其间。

苏：我还可以再问，假设，你的身体此时已经全部完蛋了，海马体、前额叶全报废了。

尼：那我肯定没戏了。

苏：不，不一定。假设，海马体和前额叶都可以被制造，在你的记忆体失灵前，它们的内容被保存到一个人造的记忆体里，然后再把这个人造的记忆体移植到另一个大脑里。同样地，手术过后，你还是你吗？

尼：你是要逼我说“我还是我”吗？那照你这个说法，生死就没有界限了，你岂不是解决了永生的难题，我们都可以永生了。

苏：这可是你说的，不是我说的。

尼：如果解决了永生的问题，那么人生痛苦的问题也自动解决了。其实怕死，更多时候是怕因死而带来的痛苦。临死时的生活，所谓往死里活（living dyingly），是最难过的。有了这招，当你痛苦时，你可以安乐死，

然后随时活过来就行了。

苏：果真如此，我们是不是应该定个标准，你不能想走就走，这毕竟是个社会问题？

尼：比利时允许安乐死，只要有三位医生和一个心理医生签字。这大概是心理医生最重要的用处，活着时用不太着。我们可以采用类似的标准。

苏：灵魂和肉体，哪个更贵？

尼：这个不好说。你走后的这些年，哲学分成了两派：您的学生柏拉图肯定说灵魂更贵，他后来的继承者都是理性主义者，他们更加信服数学而不信服物理，他们在当下流行的人工智能里，属于规则派，他们信服逻辑。另一派相信应该跟着感觉走，感觉器官是第一原则。当下正红的 DeepMind 里的一票人就是这么说的。按照我们刚才的推理，感觉器官不那么重要。

苏：你觉得灵魂和肉体，或者心-身问题解决了吗？

尼：我不知道，如果那个人造记忆体有重量的话，灵魂和肉体就应该是分离的吧？不是有个不知天高地厚的医生号称测出了灵魂的重量是 21 克吗？①

苏：哈哈，你上套了，我再问你，灵魂是永存的吗？

尼：那个人造记忆体就是灵魂，活力（elan vital，vital force）。有谣传说明斯基临死时冷冻了尸体。也许盼着某一天能暖过来把他的记忆移植到一个新的活体上。冷冻尸体成本太高，如果有技术能找到灵魂，直接冷冻灵魂就行了。

---

① 1907 年美国马萨诸塞州的医生当麦克道高尔（Duncan MacDougall）测量了 6 个濒死人死前和死后的体重差，得出结论，灵魂的重量是 21.3 克。但这个实验并不被认为是严谨的。明白人应该看出我这里引用这个实验，是个玩笑。

苏：我今天告诉你的都是可能的。你是不是感觉好多了？

尼：嗯。是不是解决了永生问题，就解决了所有问题？

苏：其实不是，永生只能解决那些给人无限时间都可以解决的问题，现在的超级计算机（supercomputer）可以算得比人快，但是如果给人无限的时间，所有超级计算机可解的问题，人也都可解。还有些问题是无论给计算机多少时间，也无法解决的问题。能解决那些问题的装置，我们暂且叫“超计算机”（hyper-computer）。我不知道永生能不能解决“超计算”的问题。

尼：你离开我们是不是为了追求“超智能”？哦，你是不是已经把你的记忆体留在哪了？

附录 1
图灵小传

Beauty is truth, truth beauty,—that is all Ye know on earth,

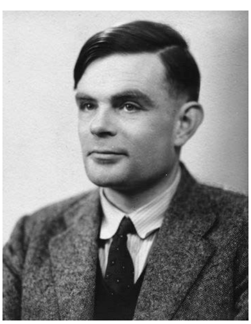
and all ye need to know.

美就是真，真就是美——你只能知道这么多，

你只需知道这么多。

——John Keats（济慈）

图灵（Alan Turing）生于 1912 年 6 月 23 日，逝于 1954 年 6 月 7 日，活了不到 42 岁。生得不伟大，死得不光荣。一方面由于他的性取向，另一方面由于他在二战中从事过机密工作，图灵在生前并没有被广泛了解。数学家兼同性恋运动活动家霍奇斯（Andrew Hodges）1983 年出版了《艾伦·图灵传：如谜的解谜者》(*Alan Turing: The Enigma*)，为我们展示了图灵谜一样的一生。该书后来一而再再而三地重版，书的内容没有本质变化，最新的一版出于 2014 年，配合好莱坞大片《模仿游戏》的发行。



图灵（1912—1954）

图灵生在西伦敦，但按日子算，他妈是在印度怀上的他。他爸是大英帝国驻印度的公务员。图灵出生的时候，他爸在印度已经工作了十年。他父母把印度当家，偶尔回英国是为了度假。图灵有一个哥哥约翰。他家祖上应该都是聪明人，他爷爷毕业于剑桥三一学院，是学数学的，他姥爷是印度马德拉斯铁路局的总工程师。他妈家有个远亲，据说是最早在理论上发现电子的人。汤姆逊六年后才在物理上证实，那远亲因此进入皇家学会，相当于中科院院士。

20 世纪初，英国流行父母把孩子留给保姆养育，不知这同罗素的教育理论是否有关系。可能是经济或其他原因，中国 20 世纪 50 年代到 70 年代，也流行过双职工把孩子交给祖父母或保姆养育的，实行独生子女政策后才改过来。图灵的保姆汤普森小姐回忆说“这孩子正直，聪明”。她的例证是图灵和别人下棋，从不让子，也不悔棋。

他三岁时，他妈到伦敦看完他又要回印度，临别时对他说：“当个乖孩子，啊！”图灵回道：“但有时我会忘的。”图灵十岁时进了预备学校，这是为了进公学做准备的。英国的公共学校（public school）相当于美国私立中学，叫 public 是因为面向社会，不是光有钱就能上，得考。预备学校和公学的主要课程是经典（指拉丁、希腊），这都不是图灵的兴趣。他喜欢智力挑战，而且一些东西自己觉得会了，就没兴趣了。他考试时，总是先挑难的做，这也是他考试总考不好的原因。他和哥哥被寄养在华德上校夫妇家，过的是乏味的生活。图灵十岁时就有厌世的情绪，自视甚高就会孤独，维特根斯坦也这德性。

他爸本以为在印度会得到提升，但上级没把他当回事，于是愤而辞职，举家搬回欧洲。图灵读的谢伯恩中学，1550 年就建校了，是英国名校，到现在还是男校。他中学成绩一般，但老师同学都注意到，这孩子上数学

课不听讲，也不看书，所有定理都是自己推出来的。如果自己推对了，考试成绩就好，自己推错了，成绩就不好。这玩意哪说理去，所有中学以前的数学知识他自己从头发明了一遍。化学课也如此，他自己发明了从海藻里分离碘，化学老师都没整明白。其他科学科目成绩也不错，但校长还是在他爸那儿告了一状："你孩子偏科，我们这儿培养文化人，他要是想当科学家，那来错地方了。" 他哪知道这孩子后来还被追认成有史以来最伟大的哲学家之一啊。

他父母为避税，定居法国。图灵兄弟俩只能从法国过英吉利海峡去各自的学校。图灵在南安普顿上岸太晚，结果所有去学校的交通都没了，于是他从行李里拿出自己的自行车，买了张地图，就向学校骑行。车太不给力，中途坏了两次。60 英里地，他走了一夜，中间还住了五星级酒店。整个一侯宝林《夜行记》啊。最后图灵把五星级酒店的发票给父母寄去，表示自己没乱花钱。

关于图灵是何时知道自己性取向的，霍奇斯作了一番考证。图灵初恋对象是中学同学摩尔康（Christopher Morcom）。同维特根斯坦喜欢苦孩子不同，图灵的恋人都智力高尚家境富有。图灵同摩尔康一起讨论科学，但他们的关系从没有超越精神。

摩尔康身材高挑，比图灵大一岁，也高一级，梦想学校是剑桥大学三一学院。这让图灵也把三一学院定为自己的目标，并给图灵带来了学习动力，他决定提前一年高考，这样可同摩尔康接着做同学。但成绩不给力，文科不行，英文全班倒数第一，拉丁文倒数第二。提前高考失败，图灵只得再努力一年。尽管头一次考试失败，但图灵回忆，他同摩尔康在剑桥一起考试时的日子是他这一生中最美好的日子。摩尔康寒假时还同图灵一起讨论科学。1930 年 2 月 6 日晚，他俩一起去听了音乐会。晚上图灵回家，

做梦惊醒，醒来的念头居然是“再见，摩尔康”。图灵不知道，那晚摩尔康病倒，一周后离世。图灵和摩尔康的故事还被写进了百老汇（其实是外百老汇）的音乐剧《电动熊的情歌》（*Lovesong of the Electric Bear*）。其中扮演摩尔康的是英国女演员博伊德（Cassidy Boyd），女扮男装。有时会想，英美的编剧怎么会对一位数学家感兴趣？布莱希特写《伽利略》时是啥心情？摩尔康的妈妈是文艺女青年，早年毕业于法国索邦，自己还有艺术画廊。摩尔康死后，图灵总去看摩尔康太太，他们成为了终身朋友。图灵觉得摩尔康仍然活着，通过自己活着。

关于图灵的一生，有两件事我不明白。其中一件是，图灵测试说的是如果把人和机器放在两个黑盒子里，如果不能区分，那么机器就是有智能的。这给人一个印象，图灵是支持“机器有智能的”。但是，图灵自己在日记里却说，机器智能其实赶不上人。这有点像数学家、哲学家普特南（Hilary Putnam），普特南一辈子一会儿支持实在论，一会儿反实在论，不知到底是哪一派的。其实图灵也类似，他提出的问题和他期望的答案不一样。他问自己的问题，写成文章问世人，别人给出了不同的答案。后人说图灵给争论的双方都提供了弹药。

另一件事是，我不明白图灵为什么只发明了图灵机，而没有发明量子图灵机。摩尔康的妈妈以摩尔康的名义给剑桥捐了个奖学金，图灵是第一个获奖者，奖项中包括一本冯诺伊曼的著作《量子力学的数学基础》。图灵在剑桥有段时间痴迷量子物理。但量子图灵机是牛津数学家、物理学家多伊奇（David Deutsch）于 1985 年提出的。关于我对图灵的这两大疑问，彭罗斯（Roger Penrose①）曾写过两本书，企图回答这两个问题，但争论颇多。

---

① 彭罗斯是霍金的数学合伙人，而霍金是牛顿的正统衣钵传人。

摩尔康死后，图灵觉得自己不会再爱上别人；他要做的无非是完成摩尔康的理想。所以，我认为我对图灵的两个困惑是摩尔康给图灵出的两个难题。图灵如愿考上了剑桥大学，三一学院录取但没给奖学金，国王学院给奖学金，图灵最后在剑桥大学国王学院学数学。摩尔康之死让图灵觉得应该追求终极的知识。他变了。

20 世纪 30 年代的国王学院人才济济，数学系有刚从牛津大学回归的哈代、纽曼，当然还有凯恩斯（也是同性恋）。他们都欣赏图灵。图灵天生内向，讲话略带结巴，嗓音尖利，不喜欢集体活动，不喜欢体育。但他很快发现了自己的长跑天分，后来得了剑桥长跑冠军。他的第一次性经历是与数学同学阿特金斯（James Atkins）。

图灵对逻辑感兴趣大约是在 1933 年，那时他读到罗素的《数理哲学导论》。1939 年图灵回剑桥大学教《数学基础》课，而同一学期维特根斯坦也在开一门同名的课程。图灵是讲数理逻辑，而老维则在讲数学哲学。图灵出于好奇，去旁听维特根斯坦的《数学基础》课。后来维特根斯坦的学生整理出版了《数学基础讲义》，这书虽然说的事很深，但是采用对话体，现代版的苏格拉底和柏拉图。其中记载的最多的对话就是老维和小图，整个课程变成了老维和小图的对掐，有意思啊。说到悖论，维特根斯坦说那只是语言游戏而已。但图灵不同意，他跟老维叫板说："如果你使用的演算里有悖论，桥会塌的。"尽管观点不同，但两人在智力上互相尊重。据维特根斯坦的传记作家蒙克（Ray Monk）的说法，当图灵不来上课时，维特根斯坦会很失望，因为他的很多思想是在与学生的对话中形成的，他一直认为哲学的教学就是对话。关于这段维特根斯坦和图灵的交流，我曾有专文论述，见《哲学评书》"维特根斯坦，图灵：1939"。特别值得指出的是，我认为维特根斯坦和图灵是最早考虑计算复杂性的人。

说牛顿是在苹果树下被掉下来的苹果砸了一下才想出万有引力，这是胡扯。图灵自己回忆他是躺在草坪上把图灵机的构造想明白的，他看到哥德尔那篇文章后就开始琢磨图灵机。哥德尔之后，大家企图在更基本的层面构造演算装置。普林斯顿大学的丘奇（Alonzo Church）发明了 λ 演算。图灵在剑桥的导师纽曼看到了丘奇的文章，就把图灵推荐给丘奇，让小图跟丘奇读博士，并告诉丘奇：这孩子搞了个图灵机。丘奇一看，推荐发表。这就是后来被大家称作"丘奇–图灵论题"的东西。简单地说，图灵机是最强的可实现的计算装置。注意这是论题，不是定理。这个论题的证据是所有已知的计算装置在可计算性上是等价的：λ 演算，Post 系统，哥德尔的递归函数，以及图灵机。按照哥德尔的说法，图灵机是最令人信服的。冯诺伊曼再次慧眼识英雄，就像他高度评价哥德尔一样，他高度评价了图灵。其实正是冯诺伊曼给这个领域起的名："可计算性"。

图灵在普林斯顿大学的两年很无聊。其实那时普林斯顿相当自由，也有同性恋活动，但他没参加过。他性格上自闭，而且不喜欢美国人的行事方式。有记载的一次体育活动还是同女校瓦萨尔学院的女生们玩了一次曲棍球。他博士一毕业，就要回英国。冯诺伊曼让他留下当助理，他婉拒了。图灵从美国回来后申请剑桥大学数学讲师职位被拒，只得接着做研究员（Fellow）。他除了继续数学研究（包括概率论、代数、分析和数论）外，开始在布莱彻利庄园兼职。布莱彻利庄园是英国政府代码和加密学校（Government Code and Cypher School，GCCS）所在地。该庄园的主要职责是为英国海陆空三军提供密码加解密服务，是机密机构。现在这儿已成为英国国家计算机博物馆的一部分。

1939 年 9 月 1 日德军占领波兰，2 日英国对德宣战，3 日图灵被召去布莱彻利庄园全职工作，负责破解德国传奇 Enigma 加密机。其实波兰早

就对德国有警觉，他们利用自己聪明的人力资源，组织数学家破解了较早版本的 Enigma。但德国人改进了。图灵在波兰人的基础上破解改进的 Enigma。图灵在紧张工作的同时爱上了同事，这回是个女的，数学家克拉克（Joan Clarke），也是搞加解密的。图灵向克拉克求婚，她答应了。图灵的诚实使得他不得不告诉她其实自己是同性恋。克拉克也忍了。但最后图灵自己不干了。电影《密码迷情》里演克拉克的是温斯莱特。

搞过密码学的都知道，加解密这东西就是道高一尺魔高一丈。你改进加密，我改进解密，不断斗争。Enigma 的主要用户是德国海军，他们的潜艇 U-boat 就是用这玩意。德国人也教会了日本人，这给美国人制造了很大的麻烦。美国人请英国帮忙，英国派出了图灵，1942 年图灵再次踏上美国领土。图灵在与 Enigma 的斗争中，逐渐形成了如何建造一台实用的通用计算机的思路。1946 年年初，他向国家物理实验室提交了 ACE 的报告，这份报告比冯诺伊曼的 EDVAC 报告晚了几个月。所以大家还是觉得老冯是最早的计算机设计师，但老冯逢人就说“这都是图灵的主意”。

图灵的长跑纪录是奥运水平的。在布莱彻利庄园工作时，经常要到伦敦开会，战时单位派车不方便，图灵也不摆谱，说一句“我自己解决吧”，64 公里路，跑着去，完事再跑着回来。战后，为了表彰破解德国密码的贡献，他被授予 OBE 爵位。足球明星贝克汉姆就被女王授过这玩意，图灵和小贝得的是低级的 OBE，不能被称为“Sir”的。图灵被授予 OBE，是为了奖励他战时的机密工作，所以颁布爵位的过程也是保密的。20 年后大家才知道图灵得过爵位。图灵本来想代表英国参加 1948 年伦敦奥运会，但他受了伤，只得放弃。图灵要是只练长跑，说不定能获更高的奖项，可能都封“Sir”了。

1948 年图灵向国家物理实验室请假，到他的母校剑桥大学国王学院

兼职研究员，表明了他重返学术的意愿。他本科时的导师纽曼这时在曼彻斯特大学数学系担任主任，他把图灵拉来做数学系的“Reader”。英国的这个“Reader”大概相当于美国的教授，王浩曾经在牛津做“Reader”，他自称“准教授”。此时曼彻斯特大学的电工系主任是威廉姆斯，他正在建当时的另一台存储程序计算机 Mark-1。当地的报纸把这台机器叫作“Electric Brain”，这大概是计算机头一次在媒体上被称为“电脑”。纽曼让图灵帮威廉姆斯做 Mark-1 的软件，但图灵此时已对工程细节失去兴趣。他在私信里表示自己的兴趣已转向“如何构造大脑的动作”。为了“电脑”这个称呼的事儿，英国公众知识分子还辩论是不是合适。反对者有著名的科学哲学家波拉尼（Polanyi）。图灵装作不知道有这回事，没参与。其实波拉尼私下和图灵是朋友，他一直催促图灵把他的想法写成文章。就是那篇著名的“计算机与智能”（Computing Machinery and Intelligence），后来发表在哲学杂志 *Mind* 上。这篇文章定义了“图灵测试”，简单地说，就是如果人不能区分放在黑箱子里的机器是人还是机器时，这台机器就该被断定为有智能。

图灵和克里斯托弗·斯特拉切（Christopher Strachey）是好朋友。克里斯托弗的叔叔立顿·斯特拉切是著名文学批评家，是罗素、凯恩斯、伍尔芙等一票人的铁哥们。克里斯托弗和后来的图灵奖获得者司考特（Dana Scott）创立了“指称语义学”，在中国有传人。图灵很早就有了计算机下棋的主意，是斯特拉切 1952 年实现了第一个跳棋程序。图灵是这个程序的第一个用户，图灵无惊无险地赢了。在人工智能第一次达特茅斯会议后的 1957 年，司马贺（Herbert Simon）断言十年内计算机国际象棋程序很快会赢人。这要到 1996 年 IBM 的“深蓝”电脑赢了当时号称要捍卫人类尊严的世界冠军卡斯帕罗夫才算数。

而“图灵测试”取得里程碑式的进展则要到 2011 年。这次还是 IBM，它的“沃森”（名字取自 IBM 的创办人沃森）超级计算机在美国的电视智力竞猜节目中击败了人类。这下大家真要想想了，这个比下棋下输的后果严重多了。在“图灵测试”后，大家在试图寻找更好的智能定义。

图灵 1951 年入选皇家学会，这是英国最高科学荣誉组织。图灵仍然同剑桥大学来往密切，他同剑桥大学国王学院的两个学生保持着亲密的友谊。两人都是学数学的，一个是约翰逊（Neville Johnson），他被证实是图灵的爱人。另一个是甘迪（Robin Gandy），必须多说几句，他是图灵的衣钵传人，图灵死后的遗物（书、信等）都交甘迪保管。甘迪也是图灵的爱人，这一八卦是一个导演先传出来的。1986 年伦敦西区出了一部说图灵的戏《破解密码》，戏中扮演图灵的是德雷克・雅可比（Derek Jacobi）爵士，后来这部戏在百老汇连演两年也很火，曾得多项托尼奖提名。甘迪被导演拖去看戏，在后台，导演神秘地介绍他说“这就是图灵的爱人”。令人不解的是，甘迪在图灵文集的出版工作上十分拖沓。1959 年图灵母亲萨拉在图灵传记序言中就宣布要出版《图灵全集》（*Collected Works of Alan Turing*）。一开始的主编是图灵的老师纽曼。纽曼 1963 年退休后，主编的职务就传给了甘迪。北荷兰出版社为此制定了专门的编辑，但每次都是甘迪掉链子。直到 1987 年，在图灵遗嘱执行人弗班克（Furbank）的干预下，才有了进展。弗班克把图灵全集分为四卷，第一卷数学，第二卷逻辑，第三卷机器智能，第四卷形态学（数学生物学）。只要求甘迪负责第二卷。一三四卷都在 1992 年出版，而甘迪负责的第二卷却迟迟没出。直到甘迪 1995 年死后，他的学生耶茨（Yates）接手才有进展，第二卷终于在 2012 年图灵 100 年诞辰时出版。

与维特根斯坦喜欢普通劳动人民不同，图灵的爱人大多是知识分子。

但图灵最后却栽在一个普通劳动人民的爱人上，这个人就是莫瑞（Arnold Murray）。图灵在一个酒吧里遇见了他，几杯酒后图灵就把他带回家了。也许人到中年，品味会变，不好才，更好色了？一个月后图灵就发现莫瑞是个贼。他一开始只是从图灵钱包里拿钱，图灵也没当回事，可能觉得人没把自己当外人呢，图灵干脆就借钱给莫瑞。但没过多久，图灵家失窃了。他知道肯定与莫瑞有关。图灵伤心地问莫瑞，莫瑞招了：有个哥们在酒吧认识的，叫哈里，有嫌疑。图灵决定到警察局报案。但此时他还想着怎么保护莫瑞，就编了一套嗑。结果一到警察局，人家已经把哈里逮住了，并且已经供出了莫瑞。警察只问了图灵一句："你和莫瑞啥关系？"答："爱人。"再问："干过啥？"图灵就把时间、地点、几次全招了……这还算干过加密算法的。

警察都没想到，本来只是逮个小偷，结果人家自己招个大的。马上以 Indecency 罪名起诉，这大概相当于中国以前的流氓罪。50 年前，王尔德（Oscar Wilde）也是以同样的罪名被起诉，援案照抄。英国这个同性恋流氓罪直到 2000 年才立法改过，和中国取消流氓罪的时间差不多。图灵那时想顶着不认罪，他哥约翰是律师，说"好汉不吃眼前亏"。图灵的律师向法庭陈述：图灵是国家的功臣，他还会为我们这个社会做贡献，把他关起来，是公共利益的损失。法官给图灵两个选择：服罪，或接受治疗。那时对付同性恋就是打荷尔蒙，图灵决定接受治疗，打雌性荷尔蒙。所有的图灵传记都说那玩意会把胸部打大的，不知他们是否在暗示什么。

图灵在给他的学生和朋友的明信片里开始夹杂自己的诗句。他常常念叨白雪公主里邪恶女王的词："吃一口苹果，像酒酿 / 让毒汁流淌。"（Dip the apple in the brew. / Let the sleeping death seep through.）一个巧合是哥德尔最喜欢的电影也是《白雪公主》，他认为只有童话才能呈现世界应有的样貌。

1954 年 6 月 8 日的凌晨，图灵被发现死于家中床上。他的床头几上有一个被咬了几口的苹果。尸体解剖说他死于氰化物中毒，死亡时间被认定是 1954 年 6 月 7 日。关于图灵的死，有三种说法：其一，事故，这是他母亲的说法，因为这孩子小时候就喜欢玩化学，不小心整错了；其二，自杀，这是目前正宗的说法；其三，阴谋论，因为图灵接触的战时机密太多了。而且那时英国、美国已经出现了几起案件：苏联用男色和女色引诱英美高级人员，无论是同性恋还是异性恋者，又值美国麦卡锡主义流行。目前关于这方面的档案，英美还不肯公开。我们还是用霍奇斯的假设：自杀。一段时间有人相信阴谋论：可能政府害怕图灵因为好色而泄露机密，所以下了黑手。其实支持阴谋论也不是没有根据，图灵的事儿一出，英国情报机构马上吊销了图灵的安全许可证（security clearance)。在美国做过网络安全工作的人都知道，没有安全许可证，基本上没办法在这个行业混。天才沃尔弗拉姆（Stephen Wolfram）的新书 *Idea Makers* 有一章专门讲图灵。他认识一些认识图灵的人，所以他更倾向于阴谋论的观点。

新西兰逻辑学家寇普兰（Jack Copeland）也是图灵专家，他 2012 年出版过一本图灵战时工作的评传。他曾根据最新的资料对图灵的最后几年做过翔实的研究，结论和霍奇斯有所不同。他认为图灵死前的两年未必像传说中那么悲惨，他有过新的恋人并和恋人一起去希腊愉快度假。“阉割治疗”对图灵智力的影响也没那么大。图灵在 1952 年开始注射雌激素时还发表了他的数学生物学的文章，现在这被认为是非线性动力学的启蒙文章。这是图灵给人类的最后的贡献。这时他的声誉已经被怀疑，已无法从事密码学和计算机研究。有证据表明，他对数学生物学的兴趣受到他的初恋摩尔康的影响，他觉得摩尔康活在自己身上。甘迪曾多次被问及图灵的死因。他每次回答时都有些激动：“有些事太深太隐私，不该被八卦。”

（Some things are too deep and private and should not be pried into.）

美国计算机学会 1966 年设立图灵奖，被称为计算机科学界的诺贝尔奖。最早赞助人是贝尔实验室，奖金只有区区几千美元。后来 Intel 接手，奖金变成了 25 万美元。2014 年谷歌加入，把奖金抬到了一百万美元。尽管苹果公司和图灵致死的苹果没关系，但他们真该赞助图灵奖。图灵死得不光荣，他的名誉都是死后得来的。2009 年 9 月 10 日图灵死后 55 年，在英国人民的强烈呼吁下，英国首相布朗向全国人民正式颁布对图灵的道歉。布朗说："我很骄傲地说，我们错了，我们应该更好地对待你。"著名数学家、逻辑学家戴维斯亲自在霍奇斯的《艾伦·图灵传：如谜的解谜者》中写过一段评论，最后一句是：对图灵的指控是悲剧，他应该被当作民族英雄。

曼彻斯特公园里图灵雕像的底座上引用了罗素的话："数学不仅有真理，也有最高的美，那是一种冷艳和简朴的美，就像雕塑。"我觉得合适。以前不懂为什么蒙克把图灵列为十二位最伟大的哲学家之一。看了罗素的话，我明白了：伟大光荣其实无所谓，重要的是"真"和"美"。

附录 2
人工智能前史：图灵与人工智能

It is humbling to read Alan Turing's papers. He thought of it all First.

阅读图灵的文章，真是令人折服。这些玩意儿，

他早就想到了，没人比他更先知。

——Rodney Brooks（布鲁克斯）

如果以“人工智能”（Artificial Intelligence）一词的最早出现时间来标志这一学科的诞生，那么它的生日差不多是 1955 年或 1956 年。一般人认为这是麦卡锡的原创，但麦老晚年承认他也是从别人那儿听来的，但想不起来是谁。事实上，英国人在此之前一直就有“机器智能”（Machine Intelligence）的说法。在维基百科上，“机器智能”这一词条被重定位到“人工智能”，也就是说这两个词儿是同义词。

图灵 1950 年在英国哲学杂志《心》（*Mind*）上发表文章“计算机与智能”，文中提出“模仿游戏”，被后人称为“图灵测试”。这篇文章被广泛认为是机器智能最早的系统化科学化论述。但图灵在 1941 年战时就开始思考机器与智能的问题，1947 年图灵在伦敦皇家天文学会就机器智能发表演讲。1948 年图灵把这次演讲整理成文章，题为“智能机器”（“Intelligent Machinery”），作为英国国家物理实验室（NPL）的内部报告，但没有公开发表。这篇文章迟至 1969 年才在年刊型论文集《机器智能》上发表。但由于和 1950 年文章的题目类似，并没有引起人们的重视。

严肃的“机器智能”的概念应该就是由图灵 1948 年在这篇 NPL 报告中最早提出的。“人工智能”的说法一直到很晚才在英国开始普及。图灵 1950 年“计算机和智能”的文章使得“机器智能”的说法被更广泛地流传。但图灵这两篇文章是前后关系，而不是并行关系。1948 年的文章对智能的概念采取了更宽泛的说法，图灵探讨了大脑皮层，他认为婴儿的大脑皮层是非组织的（unorganised）。在图灵的用语里，“非组织”就是“通用”的意思，发育的过程就是组织化的过程。他指出人身上的任何小部件都可以用机器来模仿，他还提到基因、进化和选择。正是因为如此，麻省理工学院的机器人专家布鲁克斯认为图灵（1948）是人工智能两条路线分歧的原点，而他自己的观点则是图灵 1948 年的文章比 1950 年的更为重要。

图灵 1948 年的文章提到了“肉体智能”（embodied intelligence）和“无肉体智能”（disembodied intelligence）的区分。他明确列出五个领域属于无肉体智能：(1) 博弈如下棋，(2) 语言学习，(3) 语言翻译，(4) 加密学，(5) 数学，所谓数学就是定理证明。图灵甚至提到当时的机器能处理的数学还不能涉及太多的图，也就是说一开始不适合搞几何。后来定理证明的演化很有意思，一开始重要的结果都是代数和逻辑的，但最后却是吴文俊的几何定理证明最早达到实用。一个人形机器人所需要的都属于“肉体智能”。

1948 年文章的结尾已经预示了“图灵测试”：设想 A，B，C 是三个水平一般的人类棋手，还有一台会下棋的机器。有两个房间，C 处于一个房间，而待在另一个房间的可能是 A 或机器。让 B 来做操作员，在两个房间之间传递对手的棋招。让 C 来判断另一个房间里是 A 还是机器。图灵没有再进一步说明他的目的。而在 1950 年的文章开头第一节标题就是“模仿游戏”。在“模仿游戏”中，C 是一个提问者，而一男一女 A 和 B 分别待在两个不同的房间，C 和另外两个房间的通讯只能通过打字机进行。让

C 来判别两个房间内哪个是男哪个是女。而进一步让机器分别替换 A、B 和 C，又会怎样。如果 C 不能识别房间里是人还是机器，那么机器就是有智能的。

值得指出的是，在 1948 年的文章中提出的“肉体智能”和“无肉体智能”之区分，在 1950 年的文章中，变成了“体力”和“智力”的区分，而 1950 年文章的聚焦点是“智力”。这也是布鲁克斯认为图灵 1948 年的文章更加全面的原因。图灵在撰写 1950 年的文章时已经胸有成竹，他不仅提出了问题（“机器能思维么？”），还提出了问题的各种变种，不仅给出了答案，还预想了答案的可能异议，以及对异议的反驳。图灵进一步预测到 2000 年，机器内存会达到 1GB（预测这么准还真挺神）。这篇文章为后来的一系列后学者模仿的文章提供了范文的效果，例如塞尔的“中文屋”和普特南的“缸中脑”。

二战结束后，NPL 准备研发电子计算机，图灵婉拒了剑桥大学数学讲师的 offer，于 1946 年初加入 NPL，而剑桥大学的计算机项目由刚从军队归来的物理学家威尔克斯（Maurice Wilkes）执掌。时任 NPL 头儿的是达尔文的孙子查尔斯·达尔文爵士。1946 年，达尔文爵士还在 BBC 广播节目中提到了图灵。但一年后，达尔文爵士则认为图灵的报告像是中学生作文不宜发表。图灵认为达尔文爵士和主管 ACE 计算机项目的伍默斯利（John R. Womersley）是官僚。1947 年图灵的父亲病故，他心情不好，就前往剑桥大学度假一年。但此时的剑桥大学计算机项目 EDSAC 已经在威尔克斯的全权掌控之下，图灵在剑桥大学并不受欢迎，时间不长只得离开。威尔克斯一方面企图和 EDSAC 致力仿造的美国 EDVAC 项目联系，另一方面，还主动联系图灵交流计算机的建造，1946 年年底还访问过 NPL，期望得到支持。但当图灵表达了要长久回归剑桥大学的意图后，威尔克斯感觉

到了威胁。从某种意义上，图灵是被排挤走了，一山岂容二虎。但图灵很是佩服威尔克斯的管理方式，把理论和工程的人搁在一起办公。

威尔克斯被称为“英国计算机之父”，1967 年他很不情愿地被授予了第二届图灵奖。“英国计算机之父”必须不能是“世界计算机之父”。这有点像特斯拉被 IEEE 前身的前身 AIEE 授予爱迪生奖。有意思的是，威尔克斯根本不买人工智能的账，在他的图灵奖获奖报告里，他提到，人在最早做飞机时，企图模仿鸟类，先造翅膀。但人工智能又不是仿生学，事实上，现在还真有人在做带扇动翅膀的飞机。

图灵“可计算数”[①]一文的发表日期到底是 1936 年还是 1937 年？大多数引文都用“1936”，也有一部分引文用“1937”。例如，图灵专家寇普兰（Jack Copeland）在他所有文章中都用“Turing (1936)”来指称这篇文章，而图灵传的作者霍奇斯（Andrew Hodges）大部分时间都用“Turing (1937)”。图灵的 1948 年 NPL 报告“智能机器”和 1950 年《心》杂志上那篇“计算机与智能”[②]则都用“1937”。图灵原文提交日期是 1936 年 5 月 28 日，接受日期是 1936 年 11 月 12 日。整篇文章太长，共 36 页（pp.230-265），而发表的杂志《伦敦数学学会会刊》（*Proceedings of London Mathematical Society*）把论文分成了两部分发表，第一部分（pp.230-240）发表在 vol42 part3，出版日期是 1936 年 11 月 30 日，第二部分（pp.241-265）发表在 vol42 part4，出版日期是 1936 年 12 月 23 日。发表后，逻辑学家伯奈斯（Paul Bernays）

① 这篇文章的标题全文是“论可计算的数，及其在判定问题上的一个应用”（“On Computable Numbers, with an application to the Entscheidungs problem”）。

② 这篇文章的英文标题是“Computing Machinery and Intelligence”，那时的“Computer”指的是做计算工作的人。二战期间，图灵工作的英国密码布莱彻利庄园中有很多这样的人肉计算机，多数是女性。而“计算机器”（Computing Machinery）才是我们现在说的计算机。美国计算机学会 ACM 就是 Association of Computing Machinery 的简称。

指出其中的一个小错误，图灵又写了 3 页纸的“更正”，发表于 1937 年 vol43（pp.544-546）。图灵在这篇“更正”中引用原文时用的是“1936-37”。图灵后来一直用“1937”，大概是把原文和更正加起来当作一个整体来考虑的，但也由此给后人造成了一些小困惑。

图灵在 1937 年把他 1936 年文中有关 λ 演算的思路重新整理成一篇文章，在《符号逻辑杂志》（*JSL*）上发表，在这篇文章的结尾，我们已经依稀可见麦卡锡后来定义 Lisp 语言的表述。

图灵机对实际计算机有影响吗？有两个极端派别，一派认为当然有，另一派认为一点也没有（如第二届图灵奖获得者威尔克斯）。冯诺伊曼多次向同事和部下指出 Stored-Program（所谓冯诺伊曼架构的核心）就是通用图灵机（UTM）的原创概念，应该全部归功于图灵。我个人认为，冯诺伊曼架构中真正原创的是随机存取寻址（Random Access Memory）。从这个角度看，图灵机是现代计算机的基础。倒是那些早期的计算机，从理论上看没什么价值，它们不过是巴贝奇 200 年前提出的分析机的电子实现而已。

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. TURING.

[Received 28 May, 1936.—Read 12 November, 1936.]

The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable *numbers*, it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers for explicit treatment as involving the least cumbrous technique. I hope shortly to give an account of the relations of the computable numbers, functions, and so forth to one another. This will include a development of the theory of functions of a real variable expressed in terms of computable numbers. According to my definition, a number is computable if its decimal can be written down by a machine.

In §§ 9, 10 I give some arguments with the intention of showing that the computable numbers include all numbers which could naturally be regarded as computable. In particular, I show that certain large classes of numbers are computable. They include, for instance, the real parts of all algebraic numbers, the real parts of the zeros of the Bessel functions, the numbers π, e, etc. The computable numbers do not, however, include all definable numbers, and an example is given of a definable number which is not computable.

Although the class of computable numbers is so great, and in many ways similar to the class of real numbers, it is nevertheless enumerable. In § 8 I examine certain arguments which would seem to prove the contrary. By the correct application of one of these arguments, conclusions are reached which are superficially similar to those of Gödel†. These results

† Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I", *Monatshefte Math. Phys.*, 38 (1931), 173–198.

图灵 1936 年的文章首页

附录 3
冯诺伊曼与人工智能

Talent hits a target no one else can hit; Genius hits a target no one else can see.

才华能让人击中别人击不中的目标；

而天才则能击中别人看都看不到的目标。

——Arthur Schopenhauer（叔本华）

冯诺伊曼生于 1903 年 12 月 28 日，逝于 1957 年 2 月 8 日。他在数学、理论物理和逻辑领域都做出了很多贡献。他同辈的朋友和晚一辈的同行都认为他是当时最聪明的人。冯诺伊曼 19 岁就读布达佩斯大学时，曾花大量时间去柏林听爱因斯坦的“统计力学原理”的课，只在考试时才中断。1930 年，冯诺伊曼先知先觉地加入普林斯顿大学任教，1933 年加入普林斯顿高等研究院，作为创始数学家之一。二战期间他为曼哈顿计划工作。二战结束后，他的大部分精力都花在电子计算机项目上。计算机科学一直都有两条互相交错的路线，工程路线终究可以追溯到冯诺伊曼，而理论的起源则在图灵。他们共同关注的课题是大脑和智能。



冯诺伊曼（1903—1957）

冯诺伊曼做了无数一流的工作，但他没有像哥德尔定理或图灵机一样的特级成果。以赛亚・伯林借用古希腊诗人阿基罗库斯（Archilochus）关于刺猬和狐狸的比喻，把人分为刺猬和狐狸两种，狐狸是全才，知道很多事，但刺猬只知道一件大事。全才科学家弗里曼·戴森借用类似的说法，把科学家也照此分类，但他用了飞鸟和青蛙的比喻：鸟更像是刺猬，而青蛙更像是狐狸。在戴森看来，希尔伯特、杨振宁都是高瞻远瞩的鸟，而冯诺伊曼和费曼则属接地气的青蛙。爱因斯坦当然是超级大鸟。

是冯诺伊曼发现了哥德尔定理的重要性，他称哥德尔是亚里士多德以来最伟大的逻辑学家。他曾半开玩笑地说，要是他不知道哥德尔不完全性定理的话，说不定他很快就能证明一阶逻辑是完全的呢——就在他得知哥德尔不完全性定理的前几天，他做梦证明了完全性定理。

冯诺伊曼欣赏并提携了图灵。其实，在图灵 1936 年那篇开天辟地的文章刚出来时，冯诺伊曼并没有立即意识到这篇文章的重要性。他在给图灵写奖学金推荐信时，提到了图灵在冯诺伊曼自己感兴趣的几个领域里展现的才能，却没有提及逻辑和图灵机。也许是由于他还没有从 1931 年哥德尔定理给他造成的心理冲击中缓过劲来。当时所有关注逻辑的主流数学家，在哥德尔定理之后，都与逻辑渐行渐远。倒是哥德尔最早慧眼识英雄，他一开始也没有对自己的递归函数那么有信心，但在得知图灵机的那一刻，他立即认为图灵机比自己的递归函数更令人信服。据冯诺伊曼的朋友们回忆，在 1938 年图灵回英国前，冯诺伊曼曾想把图灵留在普林斯顿做自己的助手，但已经对美国生活厌恶的图灵婉拒了。冯诺伊曼应该是在 1942 年左右认真读过图灵 1936 年的那篇文章。

冯诺伊曼在计算机工程的开创性工作是计算机产业的基础。所有的人都在受益。他牵头撰写的 EDVAC 报告定义了“冯诺伊曼”架构，后来

IBM 等所有其他计算机项目都以此为基础建造计算机。EDVAC 报告中最核心的概念是“存储程序”（Stored Program），冯诺伊曼把这个概念的原创权公正无私地给予了图灵。正像图灵专家、新西兰哲学家寇普兰（Jack Copeland）考证的，冯诺伊曼生前向他的同事多次强调，计算机中那些没有被巴贝奇预见到的概念都应该归功于图灵。所谓存储程序就是通用图灵机。EDVAC 报告真正的思想原创应该是随机寻址和其衍生品寄存器，但这个重要性和存储程序没法比。有意思的是，弗里曼・戴森的儿子乔治・戴森的一本以普林斯顿高等研究院研制最早的计算机为背景的书竟然以《图灵的大教堂》（*Turing's Cathedral*）为题，而书中的主角明显是冯诺伊曼，图灵在普林斯顿大学不过是匆匆过客。如果我们看看在谷歌 Ngram 中“图灵”和“冯诺伊曼”出现的频率，会发现图灵的逆袭出现在 20 世纪 80 年代初。



“图灵”与“冯诺伊曼”词频对比

冯诺伊曼在计算机科学界留下了无数的继承者。他 1948 年在加州理工学院所在地帕萨迪纳召开的 Hixon 会议上的演讲“自动机的通用和逻辑理论”（The General and Logical Theory of Automata）被收入他的全集，这篇文章开启了细胞自动机的理论研究。他在普林斯顿高等研究院制造计算机的助手伯克斯（Authur Burks）战后回到母校密执安大学开办了美国最早

的计算机科学系之一。伯克斯此后一生的研究都没有离开细胞自动机。可以说，受生物学启发的人工智能研究（biology-inspired）或更时髦的说法“类脑计算”的根儿在冯诺伊曼。

伯克斯培养了第一位计算机科学的博士霍兰德（John Holland），而冯诺伊曼在细胞自动机和 DNA 的工作间接影响到了霍兰德。霍兰德发明了遗传算法，他的大弟子是比他还长六岁的科德（Edgar Codd），科德因发明关系数据库得了图灵奖。霍兰德的另一位学生巴托（Andy Barto）和巴托的大弟子萨顿（Richard Sutton）发明了强化学习。通过应用强化学习，谷歌的 AlphaGo 击败了几乎所有围棋超级大师，卡内基梅隆大学的 Libratu 赢得了德州扑克大赛。

冯诺伊曼还影响了天才沃尔弗拉姆（Stephen Wolfram）。沃尔弗拉姆一直在研究细胞自动机，他的副产品是数学软件 Mathematica 和搜索引擎 Alpha。在沃尔弗拉姆的《新科学》（*A New Kind of Science*）一书中，冯诺伊曼被提及 12 次，排名第二；排名第一的是图灵，被提及 19 次。

冯诺伊曼和经济学家摩根斯顿合作的《博弈论》让另一位数学天才纳什的心灵变得更美丽，助他获得 1994 年诺贝尔经济学奖。他证明了两人零和游戏中存在 Minimax 策略，从某种意义上，这是计算机下棋经典算法 alpha-beta 的前兆。丹奇格的线性规划单纯形算法也得益于冯诺伊曼的指教。

1955 年冯诺伊曼被诊断出癌症，人们认为这和他参与曼哈顿项目受到核辐射有关。在病中，他接受了耶鲁大学西里曼讲座的邀请，但在讲座期间，他身体太虚弱了，没法到现场。1957 年他去世时，讲稿也没有完稿，1958 年才以《计算机与大脑》为名成书。商务印书馆 1965 年就出版了这本书第一版的中译本，可惜翻译得有些粗糙，而且冯诺伊曼遗孀写的序言

被莫名其妙地删除了。这本书从某种程度上预示了人工智能的发展路线。冯诺伊曼在不同场合都高度评价了图灵机和麦卡洛克–皮茨（McCulloch-Pitts）的神经网络。图灵和麦卡洛克、皮茨尽管在他们的论文中都没有互相参考，但可能彼此知道对方的工作。

《计算机与大脑》的第一部分是“计算机”，第二部分是“大脑”，但冯诺伊曼没有把这两条路线对立，他认为这是解决同一问题的两种方法。这给当下的启示是，符号派和神经派应该互相倾听互相学习而不是掐架。另外，他们都应该学学计算理论。

沃尔弗拉姆在 2003 年冯诺伊曼诞辰 100 周年时撰文纪念，在文章结尾处提到，冯诺伊曼临终前告知他唯一的孩子玛琳娜（Marina），他有一只大箱子，要在他死后 50 年时再打开。到 2007 年 2 月 8 日，冯诺伊曼逝世 50 周年时，玛琳娜把自己的儿孙们聚集在一起，打开了这只箱子。令所有人失望的是这只箱子竟然不是冯诺伊曼的。沃尔弗拉姆在当天的博客里谈及此事，但他猜测这只箱子可能和战时的秘密相关。

冯诺伊曼的女儿玛琳娜曾任尼克松的经济顾问，她说老冯平日给人的印象是外向乐观，在对苏联政策上也态度强硬，但在他极少数几个朋友的圈子内，他是个愤世嫉俗的悲观派。冯诺伊曼临终时皈依了天主教，这出乎他所有亲近朋友们的意料，因为他被认为是不可知论者。但据为他提供临终关怀的神父说，信教并没有为冯诺伊曼提供任何慰籍或解脱，他对死亡仍然充满恐惧。

冯诺伊曼的记忆超群，在他患癌时，他的弟弟麦克在病床前大声给他朗诵《浮士德》，他甚至能精确地指出麦克跳过了哪些文字。但他临死的前几天彻底丧失了记忆，对来看望他的人除了说“你好”已经说不出任何其他的话语。

冯诺伊曼被引用最多的话是："我们应该预测所有稳定的过程，控制不稳定的过程。"（All stable processes we shall predict. All unstable processes we shall control.）其实这并非是老冯的原话，而是弗里曼·戴森转述老冯 1950 年在普林斯顿的讲座的精神，那时他是多么自信啊。



冯诺伊曼和奥本海默在普林斯顿计算机前

附录 4
计算机与智能

阿兰·麦奇森·图灵

1. 模仿游戏

我在考虑这样一个问题："机器能思维么？"这要从"机器"和"思维"这两个词的定义开始。定义很容易拘泥于词汇的常规用法，但这种思路很危险。如果仅仅根据这两个词日常是如何使用的，来给它们下定义，那必然会产生这样的问题："机器能思维么？"这一问题的意义和答案就得依靠类似盖洛普民意调查这样的统计方式。这就有点荒唐了。与其如此定义，倒不如用另一个可以相对清晰无误表达的问题来取代原题。

这个新的问题可以用一个游戏来描述，我们姑且称之为"模仿游戏"。这个游戏需要三个人的参与，一男子 A，一女子 B 和一位提问人 C（男女皆可）。提问者被单独隔离在一间屋子里，见不到其他两位。游戏的目标是让提问者判断其他两位参与者中，哪位是男，哪位是女。提问者用 X 和 Y 代表另外两人，游戏结束时，他说"X 是 A，Y 是 B"或者"X 是 B，Y 是 A"。提问者可以向 A 和 B 提问，例如

C：请 X 告诉他或她头发的长度。

假设 X 就是 A，那么 A 必须作答。A 在游戏中的任务是诱导 C 做出

错误的辨识。所以他的回答可能是：“我是短发型，最长的几缕大概九英寸长。”

为了不让提问者从语气中得到提示，问题的答案应该是书面的，最好是打印的。最理想的安排是让两间屋子通过电传打印机进行通信，或者通过中介来传递问题和答案。B 在这个游戏中的任务是帮助提问者。对她来说，最好的策略可能就是诚实回答。她还可以对回答进行补充，比如：“我才是女的，别听他瞎说。”但如果男子也可做出类似的回答，那就无济于事了。

现在我们提出这样一个问题：“如果在游戏中用一台机器来取代 A，将会发生什么？”“这种情况与玩家是一男一女时相比，提问者错判的频率是否会发生变化？”我们用这些问题取代原先的问题：“机器能思维么？”

2. 对新问题的评论

有人会问“新问题的答案是什么？”以及“这个新问题值得研究么？”我们直接解答第二个问题，免去后头无止尽的循环倒推。

新问题的好处是，它在人的体力和智力之间作了相当明确的区分。没有一位工程师或化学家敢说他们能制造出和人类的皮肤一模一样的材料。当然了，将来，这可能会成为现实。即便如此，如果只是为了让“思维机器”看起来更接近人类，而为它们粉饰上人造的“鲜活外皮”，似乎意义也不大。这个新问题考虑到了这种情况，让提问者看不到、摸不到也听不到其他两位游戏参与者。这种标准还有一些其他好处，或许能通过下面的问答体会一二。

问：请以福斯桥[①]为主题，写一首十四行诗。

答：我去，别让我干这个，我压根就不会写诗。

问：34957 加 70764 等于多少？

答：（停顿大概 30 秒钟）105621。

问：你会下象棋么？

答：会啊。

问：我的王在 K1，没别的棋子了，而你只有王在 K6，车在 R1。该你走了，你走哪步？

答：（停顿 15 秒钟）车到 R8，将军。

这种问答方式适用于几乎任何一种我们希望涵盖的人类行为。如果一台机器没能在选美大赛中胜出，我们不会加以责备，正如我们不能因为一个人跑不过飞机就认为他无能。我们的游戏设定让这些天生的不足变得无关紧要。只要参与者们认为可取，他们就可以肆意吹牛，乐意把自个儿描述成多么魅力无限、强壮无比、英勇无畏都可以。反正提问者不能要求他们实地表演。

但这个游戏有一个不足会引起批评：就是机器在游戏中的处境太过不利了。如果那男的试图去冒充机器，他肯定要露馅儿的。他会因为做算术题时很慢或常常出错而立即暴露。我们不禁要问，难道机器就没法执行某些被描述为思维的操作？不过，这些操作即使是思维，与人类的思考方式也是大相径庭的。这个反对意见确实有点道理。但是我们至少可以说，如果建造的机器能够令人满意地玩好模仿游戏，那我们就不会再被这个反对意见困扰了。

我们认为在玩模仿游戏时，机器的最佳策略可能并不是模仿人的行

① 译注：Forth Bridge，1889 建于苏格兰。

为。光模仿不见得能起到多大的作用。不管怎样，本文并不想在研究这个游戏的理论问题。我们的设想是：机器在游戏中的最佳策略是像人一样自然地作答。

3. 游戏中用到的机器

在本文第一节中提出的问题，只有在确定“机器”这个词的意义后才能得到明确。自然地，我们希望在我们的机器上可以运用一切工程科技。其次，有可能工程师们制造一台能够运作的机器。但是因为机器的设计方法在很大程度上都是试验性的，所以就连它的建造者们都不能预料这台机器的工作结果。最后，我们希望机器的定义不包括正常繁衍诞生的人类。要让定义同时满足这三个条件着实不易。举个例子，你可能会要求这些工程师们都得是同一个性别，这个要求也不一定令人满意，因为通过一个人的单个皮肤细胞，培养一个完整的个体也不是说完全不可能。真能做到的话，那将是生物科技领域最具划时代意义的突破。但是我们不认为这是在“建造一台会思维的机器”。所以，我们不得不放弃“允许一切技术”这个要求。事实上，目前对“思维机器”的兴趣主要来自一种特殊的机器上，“电子计算机”或“数字计算机”。因此，我们只允许数字计算机参与我们的游戏。

乍一看，这个限制似乎过于苛刻。其实不然，为此必须先简单了解这些计算机的本质和性能。

在游戏中，只有在数字计算机表现糟糕时，才能说数字计算不符合我们的要求，正如我们为“思维”这个词定下的评判标准一样。

目前已经有许多能正常工作的数字计算机了，人们可能会问：“何不

立即做实验？这样很容易就能满足游戏的要求。同时邀请多位提问者参与游戏，统计出判断正确的概率。”对这个问题，我的简要回答是：我们所关注的，并不是所有的数字计算机或者现有的计算机，能不能玩好这个模仿游戏。我们真正要问的是：是否存在一台想象中的机器，可以在游戏中精彩发挥。当然，这仅仅是一个简要回答，下面我们将从其他角度再来考虑这个问题。

4. 数字计算机

或许可以这么解释，数字计算机旨在执行一切能由人类计算员实现的操作。人类计算员必须严格遵守规则，在任何细节上，都不容许有一丝丝偏离。我们可以假设，有这么一本规则手册，每次有新的任务分配给计算员时，规则也会随之改变。人类计算员用稿纸做计算，稿纸的供应是不限量的。同时他还可以借助“台式计算机”进行四则运算，但这并不重要。

如果我们用上一段的解释，作为数字计算机的定义，那可能会陷入论证循环的误区。为了防止这种情况发生，我们把实现预期效果的途径列了出来。一台数字计算机通常由以下三个部分组成：

(i) 存储器

(ii) 运算器

(iii) 控制器

存储器用来储存信息，相当于人类计算员的纸，无论这是计算过程中用到的纸，还是用来印刷规则手册的纸。还有一部分存储器相当于计算员的记忆，记录计算员大脑中的运算。

运算器负责计算中的各种独立操作。这些独立操作是什么，就因机

器而异了。通常，机器可以执行一些相当长的操作，例如，“3540675445 乘以 7076345687”。但是有一些机器只能执行非常简单的操作，比如“写下 0”。

我们提到，计算员的“规则手册”占用了机器中的一部分存储器，我们不妨称之为“指令列表”。控制器的职能就是保证这些指令按照正确的顺序得到正确的执行。控制器的设计要确保这一点。

存储器里的信息通常会被分解为许多比较小的数据块。例如，一个机器中，一个数据块可能由 10 个十进制数组成。这些数字会以某种系统化的方式，被分配到多个存放在存储器上的数据块中。一个典型的指令可以这样描述：

“把存放在地址 6890 的数与存放在地址 4302 的数相加，并把结果存入后一个存储位置。”

机器中的指令肯定不是用英语标示的。这个指令更有可能是用代码的形式表达，例如 6809430217。“17”在这里指“这两个数字的多种可能操作中，会执行哪一个”。这个例子中执行的就是“加法操作……”可以注意到，这个指令总共 10 个字，正好可以放在一个存储空间中，很方便。通常情况下，控制器都能保证指令按照它们的存储顺序被执行，但是偶尔也会碰上这样的指令：

“现在执行存储在地址 5606 的指令，并从那里继续执行。”

或者这样的指令：

“如果地址 4505 包含“0”，那么下一步执行存储在地址 6707 的指令，否则继续。”

后面这种指令很重要，它使一组指令得以不断反复执行，直到某种条件得到满足。但是，要达到这个目的，就得重复执行相同的指令，不能

变。拿日常生活中的一件小事打个比方：假设妈妈希望小汤米（Tommy）每天上学时都去修鞋匠那里，看看她的鞋修好没。那妈妈可以每天都跟小汤米说一遍这事。另一个办法，妈妈可以在小汤姆每天上学都能看到的地方贴个便条，提醒小汤姆每天到鞋匠那里去看一下，直到小汤姆拿回她的鞋，撕掉那个便条。

读者必须接受一个事实，那就是数字计算机的建造是可能的，而且也正是按照我们如上描述的原则建造的。事实上，这些数字计算机已经可以非常逼真地模仿一位人类计算员的行为了。

人类计算员须要遵守的规则手册，事实上仅仅是一个虚构的设想。真正的计算员记得住他们得做些什么。如果想让一台机器模仿计算员执行复杂的操作，那必须先问问计算员，这操作是如何做到的，再把结果翻译成指令表。这种构造指令表的行为通常被称为“编程”。“给一台机器编程使之执行操作 A”，意思就是把合适的指令表放入机器使它能够执行操作 A。

上述数字计算机概念的一个有趣的变种是“带有随机元素的数字计算机”。这些指令中包括掷色子的指令，或者某种同类的电子过程。举个例子，“掷一下色子，并把结果存入地址 1000”。有时候人们认为这样的机器是有自由意志的（我自个儿不会这么说）。一般情况下，我们没办法只通过观察就能判断出一个机器有没有随机元素。如果我们从 π 小数点后面的数字中随意选择，也能产生相似的效果。

实际的数字计算机的存储空间都是有限的。理论上说，让一台计算机获得无限的存储空间并不困难。当然了，任何时候，我们只是使用有限的部分。真正能够被建造的容量也是有限的。不过可以想象，我们能够根据要求，添加越来越多的存储空间。这种计算机只有特定的理论价值，我

们称之为无限容量计算机。

有关数字计算机的设想，其实很早就有了。1828 年到 1839 年期间，担任剑桥大学卢卡斯数学讲席教授的查尔斯・巴贝奇就曾设想过这样的机器，他称之为分析机（Analytical Engine）。但是这台机器没有能被实际建成。尽管巴贝奇已经理清了所有的基本思路，但是在那个时代，他的机器并没有被看好。分析机能够达到的运算速度肯定比一个计算员要快得多，但还是比曼彻斯特机要慢 100 倍，而曼彻斯特机在现代计算机中已经算是比较慢的了。值得一提的是，分析机的存储全靠齿轮和卡片，是完全机械的。

知道巴贝奇的分析机能够完全通过机械手段建造，帮助我们破除了一个迷信。以往，我们只重视“现代数字计算机就是电的”这个观点，而且神经系统也是与电相关的。可是巴贝奇的分析机却是完全机械的。我们知道所有的数字计算机在某种意义上都是等同的。所以在理论上，电的使用其实没那么重要。当然，涉及快速发送信号时，电还是必要的。因此电在计算机和神经系统这两个方面的使用就不足为奇了。在神经系统中，化学现象最起码和电也是同样重要的。还有一些计算机的存储器还是基于声学原理。如此看来，计算机和神经系统都使用电，仅仅是表面上的共同点罢了。如果我们真想寻找这样的相似点，倒不如看看功能的数学性质。

5. 数字计算机的通用性

上一节中提出的数字计算机可以被归类为“离散状态机”。这类机器可以实现从一个确定状态向另一种确定状态的跳转。这些状态的差异之大，几乎不可能造成状态之间的混淆。但严格地说，这样的机器是不存在的。因为万物的移动变化，实际上都是连续的。但是在某些方面，一些机

器仍可以被视为离散状态机。例如照明系统中的开关，我们可以简单地把开关看成只有开和关两种状态。两种状态之间必定有转换过程，但是在绝大多数情况下可以忽略。举一个离散状态机的例子。想象一下，有一个每秒转动 120 度的轮子，一个外置的杠杆，轮子可能会受杠杆的阻挡而停止转动，并且在轮子的某个位置装有一盏灯。我们可以这样描述这个机器：机器的内部状态（通过轮子的位置来描述）可以设定为 q_1，q_2 和 q_3。输入信号是 i_0 或 i_1（控制杆的位置）。任何时间的内部状态都是由前一次的状态和输入信号按照指令表共同决定：

		上一状态		
		q_1	q_2	q_3
输入	i_0	q_2	q_3	q_1
	i_1	q_1	q_2	q_3

输出信号是唯一能够在外部进行观测的内部状态指示器（比如灯），可以用下表描述：

状态	q_1	q_2	q_3
输出	o_0	o_0	o_1

这是一个典型的离散状态机的例子。只要状态数量是有限的，就可以用这样的表格描述它们。

可以看出，只要确定初始状态和输入信号，所有的未来状态都是可以预测的。拉普拉斯有这么个观点：只要确定某一时刻宇宙中所有粒子的位置和速度，就能够预知未来的所有状态。和拉普拉斯相比，我们的预测要接近实际。在“全宇宙整体”的系统中，初始状态下一个微小的误差，可能会在将来产生巨大的影响。例如说，在某个时刻，一个电子出现十亿

分之一厘米的位置偏移，这可能将决定一年后某个人在一场雪崩中的生死。我们称为“离散状态机”的机械系统并不会发生这样的现象，这是它们的基本属性。且不说理想化机器，即使是一台实际的物理机器，只要掌握某个时刻合理准确的状态信息，也可以精确地预测其未来的一切状态。

我们所说的数字计算机属于离散状态机。但这样的机器所能够达到的状态数量是巨大的。例如，现在在曼彻斯特工作的机器大概有 2^{165000} 种状态，也就是 10^{50000} 种。而在上文轮子的例子中，只有 3 种状态。不难解释为何状态数量如此庞大。计算机中有一个存储器，相当于计算员运算用的纸。记录在计算员的稿纸上的所有符号组合，都必须能够写入存储器中。为了简单起见，假设仅将 0 到 9 这 10 个数字作为符号，忽略手写体的差别。假如计算机的存储空间相当于 100 张纸，每张纸有 50 行，每行 30 位这么大。那么这台计算机的状态数量就会是 $10^{100\times50\times30}$，即 10^{150000}，相当于 3 个曼彻斯特机状态的总和。状态数的基为 2 的对数通常被称为机器的“存储容量”。因此曼彻斯特机的存储容量大概是 165000 位，而上文中轮子的存储大小大约是 1.6 位。如果两台机器加在一起，那么合成机器的存储容量应该是这两台机器存储容量相加的总和。因此我们说：“曼彻斯特机包含有 64 个磁带存储器，每个的存储容量是 2560，此外还有 8 支电子管，每支的存储容量为 1280 位。各种各样的存储器加在一起大约是 300 个，存储容量达到 174380。”

只要给出与离散状态机对应的表格，就有可能预测机器将会做什么。这样的计算当然也能够通过数字计算机进行。只要执行的速度够快，数字计算机就能够模拟任何离散状态机的行为。如果让我们研究的机器和进行模仿的数字计算机分别充当模仿游戏中 B 与 A，提问者将难以区分它们。当然，前提是数字计算机必须有足够大的存储空间和足够快的运行速度，

而且它在模仿不同的机器之前必须被重新编程。

数字计算机因为可以模拟任意一个离散状态机器，这个特殊性质被称为“通用机器”。如果不考虑速度的话，这种机器的一个重要优点就是我们不必为不同的计算过程设计不同的新机器了。一台数字计算机就可以解决问题。我们需要做的是根据情况进行相应的编程。可以看出，所有的数字计算机在某种意义上是相互等价的。

现在，我们可以重新考虑一下在第三节末尾提出的问题。我们暂且用问题“在我们的想象中，是否存在可以在模仿游戏中表现出色的数字计算机？”来代替问题“机器能够思维么？”。如果各位不介意的话，我们再问一个貌似更加普遍的问题：“是否存在能够表现良好的离散状态机？”考虑到通用性，我们可以看出这两个问题事实上都等价于这个问题：“让我们把注意力集中在一台特定的数字计算机 C 上。我们为 C 提供足够大的存储空间和足够快的计算速度，并且对它进行适当的编程。在游戏中，由人扮演 B 的角色，那么 C 能否令人满意地扮演 A 的角色？”

6. 主要问题的反方观点

现在一切准备工作已经就绪，我们可以就“机器能思维吗？”这一问题和上一节末尾所说的这个问题的变种展开辩论了。我们尚不能全盘摒弃原题。因为仍有一些人不认同用变种来替换原题。我们至少要听听不同的意见。

我想先就这个问题阐述一下自己的看法，希望可以帮助读者简化问题。首先，我们需要换个角度，更为精准地思考这个问题。我认为在未来 50 年的时间里，计算机的信息存储量将达到大约 10^9，那时我们将能编写

出计算机程序，使得计算机在模仿游戏中如鱼得水，标准是在提问 5 分钟后，一般的提问者能准确鉴别的概率不会超过 70%。我认为原题“机器能思维么？”没什么意义，也不值得讨论[①]。到 20 世纪末，词汇的用法和普遍的成见都会发生较大的改变。那时候人们再谈论机器会思维时，就不会有如此的反对意见了。并且，我还认为，掩盖这些信念决不会带来任何益处。人们普遍认为，科学家们总是雷打不动地从可靠的事实到可靠的事实，从来不受任何未经验证的假设的影响。这种看法是相当错误的。假如能清楚地划分哪些是事实，而哪些又是未经验证的假设，那也没有什么坏处。假设非常重要，因为它们能够给研究指明道路。

我现在谈谈与我的看法相对立的观点。

(1) 来自神学的反对意见

思维是人类不朽灵魂的一项功能。上帝只赋予每个善男信女不朽的灵魂，但从未将之赐予任何其他的动物或机器。所以，动物或者机器不能思维。

尽管我不接受这种看法，但我可以试着用神学的说法来回应。我认为，如果将动物和人划为一个类别，这个观点会更有说服力些。在我看来，生物与非生物之间的差别远远大于人和其他动物之间的差别。如果把这一正统神学观点放在其他宗教团体里，就会显得更明晰。我认为，上述的论点隐含着对上帝无边神力的严重限制。必须承认，有些事情上帝也是无能为力的，比如，他不能让 1 等于 2。但是，我们难道不应该相信，要是上帝愿意，他完全可以赋予一头大象灵魂吗？大家可能指望着，上帝能发挥无上神力造就基因变种，赐予大象一个升级版大脑，满足大象灵魂的需求。

---

① 译注：原文为 The original question, “Can machines think?” I believe to be too meaningless to deserve discussion.

换成机器，我们也可以用同样的方式来论证。也许这看上去有点怪，不过是因为我们向前走得更远了一步。这其实只能说明，在这些情况下，上帝认为授予灵魂不合适。这些问题我们将在本文的其余部分再做讨论。话说回来，即使要制造这样的机器，我们也不应该傲慢地篡夺上帝创造灵魂的权力，就像人类不应该被剥夺生儿育女的权力一样。在这两种情况下，我们都只是上帝意志的工具，是他所创造的灵魂的居所。

但这仅仅是猜想而已。不管用神学论据来证明什么，我都不为所动。这样的论据在历史面前早已是漏洞百出。伽利略时代，有人企图用圣经中的“太阳在空中巍然不动……慢条斯理，不愿西落，已有一整日”（《约书亚书》，10:13）以及“他为大地奠定基础，叫它永远不动摇”（《诗篇》，104:5）作为论据来驳斥哥白尼的理论。用今天的知识来看，这种论据站不住脚。但当时没有知识，情况自然大不同。

(2)“鸵鸟”式的异议

机器思维会带来可怕的后果。但愿机器永远不会思考。

这种观点不如神学的说法那样直言不讳，但它还是影响了大多数人的想法。我们都愿意相信，在某些微妙的方面，人类比其他任何生物都更优越。这点要是能被证明，那再好不过了，因为人类就能稳居高高在上的统治地位了。神学的论点之所以这样风靡，很明显与这种情绪密切相关。这种观点尤其受知识分子的追捧，因为他们比其他人更看重思维能力，更信奉人类在这一方面的优越性。

但我认为这个论点不够充分，无须一驳。安慰倒是更须要的，这兴许能在灵魂轮回说中寻得些吧。

(3) 来自数学的异议

数理逻辑的一些结论可以用来证明离散状态机的能力是有限的。这

些结论中最著名的是哥德尔定理。哥德尔定理（1931）声称，在任何一个足够强的逻辑系统里，都能形成一个陈述，这个陈述在本系统内无法被证明也无法被否证，除非这个系统本身就是不一致的。丘奇（1936）、克利尼（1935）、罗瑟和图灵（1937）等人也有同哥德尔定理很相似的结论。图灵的结果最方便，因为这个结果直接涉及机器，而其他人的结果只能用作相对间接的论断。比方说，要使用哥德尔定理，我们除了要通过机器来描述逻辑系统，还得再反过来，通过逻辑系统来描述机器。图灵讨论的机器是一台通用的数字计算机，已经证明存在着这样的机器不能干的事情。如果把这个结论套到模仿游戏中，那么就会存在某些问题。计算机要么回答错误，要么根本回答不了，无论给予多长时间。这类问题有很多，有些问题某台机器回答不了，或许换另一台机器就能满意地回答。我们假定，现在提的都是只需要回答“是”或者“不是”的问题，不会出现像“你认为毕加索怎么样？”之类的问题。我们知道的机器回答不了的问题都是如下类型的问题：“这台机器有以下特点……那么这台机器会对任何问题作出‘是’的回答吗？”这个问题的省略号部分是对某台机器的标准描述，就像是第五节中使用的机器。如果所描述的机器与被提问的机器具有某种相对简单的关联，可以证明，要么是答案错了，要么就是压根儿没有答案。这是数学结果：这个结论认定机器能力有限，而人类智能不存在这种局限性。

对于这个论点，我的回应是：尽管已经证明任意一台特定机器的能力都是有限的，但它没有任何证据表明，人类的智慧就没有这种局限性。我认为不应该轻易忽视这个论点。每当一台机器恰好碰上一个关键问题，并给出我们确定不是正解的答案时，这会让我们产生一种优越感。这种优越感是错觉吗？显然这种优越感是真实的，但我认为这没有多大的意义。

我们人类自己平时也经常对问题作出错误的回答，没必要因为机器犯了错误就沾沾自喜。而且，优越感只是在小赢一台机器时体会到，而同时应对所有的机器，就不是那么回事了。简而言之，一个人有可能比一台特定的机器聪明，但也有可能就不如另一台机器机智。

我认为，大部分持数学异见的人，是可以接受把模仿游戏作为讨论基点的。而持前两种反对意见的人对判断标准问题估计根本就不会感兴趣。

(4) 来自意识的论断

杰弗逊教授在 1949 年的李斯特演说中，很明确地阐述了这个论点。我摘引了其中的一段话："只有在机器能够凭借思想与情感，创作出一首十四行诗或一支协奏曲，而不只是符号的随机拼凑时，我们才会认同机器与大脑是一样的。也就是说，机器不仅要能创作出来，而且要意识到是它自己创作的。任何机制都感觉不到（不仅仅是人工信号或简单装置）成功的喜悦，也不会因为困难而郁郁寡欢，因为阿谀奉承而沾沾自喜，因为犯错误而闷闷不乐，因为性爱而神魂颠倒，也不会因为事与愿违而暴跳如雷或一蹶不振。"

这个论点似乎否定了我们测试的有效性。按照这种观点最极端的形式来说，你要肯定一台机器是否能够思考，唯一的途径就是成为那台机器，并且去感受这台机器的思维活动。只有这样，他才能够向众人描述这台机器的感觉。当然，没有人会知道他讲的是不是实话。同样地，依照这个观点，要想知道某一个人会不会思维，唯一的途径就是成为这个人。这实际上是唯我论的观点。这个观点也许是符合逻辑的，但是同时，沟通交流会变得极其困难。A 可能会认为，"A 会思维，而 B 不会"。而 B 则会说，"B 会思维，而 A 不会"。我们犯不着为这个观点争执不休，不如客客气气地

默认大家都会思维。

我肯定杰弗逊教授不愿意采纳这样一个极端的唯我论观点。他可能更情愿把模仿游戏当作一个测试。这个游戏（省略了游戏者 B）在实践中经常采用“口试”形式，来鉴定某人就某事是否真的理解了，或者只是在“鹦鹉学舌”。让我们听一听这种“口试”是如何进行的：

提问者：你的十四行诗的第一行是这样的，“能否把你比作夏日璀璨？”[①]要是将“夏日”改成“春晓”，是不是也可以，或许会更好？

证人：这样一改就不合韵律了呀。

提问者：改为“冬天”如何？这样也会合韵。

证人：是没问题，但是有谁愿意被比作冬天呢？

提问者：匹克威克先生会使你想起圣诞节吗？

证人：或多或少会吧。

提问者：圣诞节不就在冬天吗，我认为匹克威克先生不会介意这个比喻的。

证人：你在逗我吗？冬天的意思是指某一个典型的冬天，而不是像圣诞节那样特殊的日子。

如果那台写十四行诗的机器在这场“口试”中能够这样回答，杰弗逊教授会作何感想呢？我不知道他是不是会认为，机器作答“只是在发送人工信号”。但是，如果机器的回答能够像上面引述的那样合情合理，连贯一致，我相信杰弗逊教授会改变“机器只是‘一项简单装置’”的看法。我认为这种说法无非是说可以把一个人诵读十四行诗的录音放到机器里，随时按个按钮就能听到这段录音。

---

① 译注：此处图灵引用了莎士比亚十四行诗中的句子 “Shall I compare thee to a summer’s day”。

综上所述，我认为大部分支持来自意识的论点的人，宁可在劝说下放弃原来的主张，也不愿陷入唯我论的困境。这些人很有可能愿意接受我们的测试。

我并不想给大家留下这样一个印象，即我认为意识并没有什么神秘之处。意识确实有神秘之处，比如，要想确定意识的物理位置，就是一个悖论。但是，即使我们没有解决这些谜，还是能够回答那个本文关心的问题的。

(5) 来自种种能力限制的论断

这些论断一般是这么说的：“即使你可以让机器做任何你刚才提到的事情，可你永远也不能使一台机器做 x。”这类 x 行为有许多特征。我在这里引用一些例子：

要善良、足智多谋、美丽、友好、干劲十足、富于幽默感，明辨是非，会犯错，会恋爱，喜欢草莓和奶油，能使别人陷入情网，会吸取经验，措词得当，自省，像人一样有多姿多彩的行为方式，乐于挑战新鲜事物。

这些话一般都用不着证明。我认为这些话都是以科学归纳的原则为基础的。一个人在一生中见过成百上千台机器，然后由所见所闻得出了一些结论。例如说，它们形态丑陋，应用范围狭窄，只要范围略有变动，它们就招架不住，并且它们的行为方式也非常单调，等等。他很自然地认为，机器通常都是这副德行。大部分机器的能力限制，归结于机器存储量太小（我们设想，存储量这个概念扩展到其他机器，不仅仅是离散状态机。精确的定义在此不重要，因为目前的讨论不要求数学上准确。）几年前，数字计算机还不为人知，要是你光说其特征不提其构造，那么人们就要说你是在信口开河。这也是人们运用科学归纳法的结果。当然，人们在使用这个原则时，大都是无意识的。一个小孩一朝被火烫，十年怕烛台，我认为

他这就是在使用科学归纳（当然，我也可以用许多别的方式来解释这一现象）。把科学归纳法用到人类的工作和习惯上，恐怕不太合适。你得研究大量的时空，才能得出靠谱的结果。否则，我们可能会（就像许多说英语的儿童那样）以为世界上所有的人都应该讲英语，学法语真是傻透了。

刚才提到的许多能力限制，我还要特别多说几句。机器享受不了草莓奶油的这种说法，可能让读者觉得有点轻率。我们或许能够做到让机器享受这些美味，机器可以被强迫去享受，但如果也企图强迫人这么做就愚蠢了。真正重要的是，这种能力的限制可能会带来其他方面的能力限制，比如，人与机器之间能形成友谊，那种像白人与白人之间，或黑人与黑人之间的友谊。

还有人认为“机器不该出错”，这种说法有点令人费解。我们不禁要反问：“如果它们出错，是不是就更劣等？”那我们试试用同情的态度，看看这究竟是什么意思。我想我们可以借助模仿游戏来解释这种说法。有人提出，提问者可以通过算术题来分辨是机器还是人，因为机器在回答算术题时总是丝毫不差，所以机器的面纱很容易揭开。这些（带模仿游戏程序的）机器并不一定非得答对所有算术题。它会故意算错，来蒙骗提问者。至于算术中的错误，还有一种可能性是机械故障，这种批评的解释也不够有同情心。限于篇幅，我们对这个问题不再做更深入的探究了。在我看来，这个观点的根源是因为混淆了两个不同性质的错误。我们称这两个错误为“功能性错误”和“结论性错误”。功能错误是由某些机械或电器故障引起的，这些故障导致机器不能够按照指令工作。在进行哲学讨论时，我们很容易忽视发生这种错误的可能性。这个时候，我们实际上在谈论“抽象的机器”。这些机器是数学的虚构，而不是物理实在。按照定义，抽象机器不会犯错，正是在这个意义上，我们说：“机器不会出差错。”当机器的输

出信号附加了某种意义时，机器才可能会出现结论性错误。例如，机器能够打印数学方程或英语句子。当机器打出一个错误命题时，我们就认为这台机器犯了结论性错误。很明显，我找不到丝毫的理由证明，机器从不犯这类错误。一台机器有可能别的什么也不会做，只会连续打出“$0=1$”。一个更合理的例子：机器会通过科学归纳来得出结论。这种办法肯定会导致错误的结论。

有种说法是机器不能成为它自己思维的主体。如果我们能证明机器的某些思维带有主体的话，就能驳回这种说法。“一台机器运行的主体”确实有意义，至少对于研究它的人来说是这样的。比如，如果一台机器试图解方程式 $x^2-40x-11=0$。我们会认为，在这个时刻，这个方程式本身就是机器的主体的一部分。从这层意义上说，机器无疑是能够成为它自己思维的主体的。这对机器编排自己的程序和预测因本身结构变化所带来的影响都会有所帮助。机器就能通过观察自己的行为带来的结果，去修改自己的程序，从而更有效地达到某些目的。这不是乌托邦式的空想，而是在不远的将来有可能实现的事。

有人反对说，机器的行为比较单一。其实就是想表达，机器无法拥有足够大的存储容量。要知道，直到最近，1000 字的存储量甚至都很罕见。

我们在这里考虑的这些异见，大多是来自意识的异议，只是稍作乔装。通常情况下，如果一个人坚持一台机器有能力完成其中的某件事，并且描述机器可能采用的方法，那么这个人并不会给别人留下多深的印象。人们会认为机器所使用的方法（不管是什么方法，总是机械性的）实在太低级了。大家可以参考前文引述的杰弗逊的一段话。[①]

① 译注：原文杰弗逊引文出现在第 22 页。

(6) 拉芙蕾丝伯爵夫人[①]的异议

关于巴贝奇的分析机，最详尽的信息来自拉芙蕾丝夫人的笔记。她这样写道："分析机没有意图想要*原创*什么东西。它能做*我们知道该怎样去指挥它去做的任何事情*。"（斜体为她本人所加）哈特里（Hartree 1949）引用了这段话，并补充道："这不是说没办法构造会'独立思考'的电子设备，或用生物学的话说，我们能够建立一种条件反射作为'学习'的基础。从最近的发展情况来看，这一设想在原则上是不是有实现的可能，已经引起了极大兴趣和关注。但是，当下的任何一台机器，不论是做好的还是没做好的，似乎并不具备这些特点。"

在这点上我完全同意哈特里的看法，我们注意到，他并没有断言当时的机器还不具备这个特点，而是指出了，拉芙蕾丝夫人当时所有的证据还不足以使她自己信服。从某种意义上讲，我们讨论的机器很有可能已经具备了这个特点。因为我们认为某些离散机器是具备的。而分析机实际上是一台通用数字计算机。因此，如果它的存储容量和速度达到一定水准，我们就能通过适当的程序让它模仿我们所讨论的机器。也许伯爵夫人或巴贝奇都没有想到这一点。但无论在什么情况下，我们不能要求他们想出所有能够被想出的东西。

① 译注：英国诗人拜伦的女儿艾达（Ada）19 岁嫁给威廉·拉芙蕾丝（William Lovelace）伯爵，故得名。1840 年，巴贝奇被邀在意大利都灵大学（University of Turin）演讲分析机，台下有个主攻工程和数学的意大利学生 Luigi Menabrea 把巴贝奇的演讲用法文做了记录。Menabrea 后来做了意大利总理。艾达是巴贝奇晚年的助手，1842 年~1843 年，她把 Menabrea 的听课笔记从法文翻译成英文。译文中，她附加了许多注记，她的注记比原来的笔记要长三倍，其中一段详细说明了在分析机上计算伯努利数的方式，这被称作世界第一个计算机程序。美国国防部 1980 年把当时新研制的编程语言命名为 Ada。她的文章还提出分析机的能力不仅仅是计算，这些已超越巴贝奇原来的构想。她的"诗性科学"（poetical science）的想法视分析机为人类的工具。天才沃尔弗拉姆的新书 *Idea Makers* 记录了 15 位他敬仰的人的生平和贡献，其中艾达的那篇最长。

我们会在“学习机器”那一节继续探讨这个问题。

拉芙蕾丝夫人异议的另一种说法是，机器“永远不会创新”。我们可以用一句谚语抵挡一下这种说法：“日光之下，再无新事（普天之下，莫非旧闻）”。人们的“原创作品”难道不是接受教育的结果？或者因为遵循已知的通用法则？这谁也不能保证。这种异议还有另一个稍微委婉一点的说法，即一台机器永远无法“让我们吃惊”。这种说法是一个更直接的质疑，对此我可以针锋相对地反驳：机器经常让我吃惊。这主要是由于我对机器能做什么估算不足。确切地说，我做的估算总是又匆忙又马虎，甚至还有点冒险。我也许会这样对自己说：“我猜这里的电压与那里的应该差不多；管他的，就当是一样吧。”我自然经常出错，但结果却往往让我大吃一惊。因为当实验完成的时候，那些假设早被我忘得一干二净了。我欢迎人们批评我的错误，但在我证实我所经历的“惊讶”时，人们也别怀疑我的诚信。

我并不奢求我的回答能让异议者们心服口服。人们也许会这样说，所谓的大吃一惊都是因为我自己有丰富的想象力，与机器本身毫不相干。如此，我们又会重新回到来自意识的那个论证上去，而背离“机器能否使人吃惊”这个话题了。这种论证已经结束了。也许还值得一提的是，要鉴定某事是否会使人惊奇，还真少不了“想象力丰富的心理活动”，甭管这个令人吃惊的事件是起于一个人、一本书、一台机器或是任何别的东西。

我认为“机器不会令人吃惊”的这个观点，是由一个错误引起的，哲学家们和数学家们都特别容易犯这个错误。它是这样一个假设，一旦某个事实出现在人的意识中，那么由这个事实引起的一切都会同时涌入人的意识。在许多情况下，这种假设十分奏效。但是，人们太轻易就忘记，这其实是个错误的假设。这样做必然会导致大家认为，靠数据和普遍原则得

出结论这样的工作没有价值。

(7) 来自神经系统连续性的论证

神经系统不是离散状态机。一个神经元承接的一个小的脉冲的错误信息会导致会导致传出脉冲极大的误差。这样一来，由此论断：用一个离散状态系统去模仿神经系统的行为是无法实现的。

离散状态机器和连续机器肯定是很不一样的。但是，如果我们严格遵循模仿游戏的条件，提问者就甭想从这个差异中占到什么便宜。假使我们能考查另一些简单的连续机器，情况就会变得更明白。一台微分分析机（A differential analyser，是一种用作非离散状态计算的机器）就完全够用了。有些这样的机器可以将答案打印出来，所以适合参加模仿游戏。要一台数字计算机精确地猜中微分分析机会如何回答一个问题是不可能的，但它是完全有能力正确作答。比如说，如果你要它回答 π 的值是多少（约等于 3.1416），它就会在 3.12、3.13、3.14、3.15、3.16 之间随机选择，选择概率依次分别为（比方说）0.05、0.15、0.55、0.19、0.06。这样的话，提问者就很难分辨哪个是微分分析机，哪个是数字计算机。

(8) 来自行为不规范性的论断

我们不可能总结出一套规则，囊括一个人在可想象的环境中的一切行为。比方说，我们可以设立一条规则：红灯停、绿灯行。但万一因为某种故障，红绿灯同时亮了呢？为了安全起见，我们也许会决定在红绿灯同时亮的时候止步。但后续还会出现很多其他问题。要总结出一套覆盖所有可能性的规则，哪怕只是关于红绿灯的规则，都是不可能的。这些看法我完全赞同。

因此人们争论说我们不可能成为机器。我试图重新论证我的观点，但老担心处理不好。或许可以这么说：“如果每一个人都有一全套行动规

则来制约他的生活，那么，人同机器就相差无几了。但实际上不存在这些规则，因此，人不可能成为机器。”这里，不周延的中项十分刺眼。我想平常没有人会这样去做论证，但我相信在这里实际上用的就是这样的论证。有人将“行事规则”和“行为规律”混为一谈，因此使这个问题有点模糊不清。所谓“行事规则”，我指的是像“红灯停”这样的规则。对这类规则你能有意识到地遵守。而所谓“行为规律”，是指自然规律。拿我们自己身体来举例的话，就像“如果你拧他一下，他就会叫唤”这样的规律。在上面的论证中，如果把“制约他的生活的行为规律”改为“他用以制约自己生活的行为准则”，那么，这个论证的不周延中项就不会那么难以克服了。因为我们相信，用行为规律制约生活，意味着人在某种层面上就是机器（虽然不一定就是离散状态机器）。反过来说，作为一台机器也就意味着制约于这些规律。然而，我们很难像否定全套的行为规则那样，轻易地否认全部的行为规律。只有通过科学的观察，才能发现这些规律。无论在何种情况下，我们都不会说：“我们做了足够的研究了，不存在这种规律。”

我们可以找到有力的证据，证明这种说法是不正确的。假定，如果这些规律存在，我们就能够找到它们。就一台离散状态机器而言，我们应该可以在一个合理的期限，比方说在 1000 年之内，通过大量的观察找到规律并预测其未来的行为。其实不然，我曾在曼彻斯特计算机内输入了一个程序，仅仅用了 1000 个存储单元，给这台机器输入一个 16 位的数，机器在两秒钟内输出另一个数。我不相信有谁可以仅仅靠这些输出就能了解这个程序，对一个没试过的输入预测可能的输出。

(9) 来自超感官知觉的论断

我想读者都熟悉超感官知觉的说法，它有四种方式，即心灵感应、

超视觉、先知先觉和意念运动。这些令人不安的现象似乎在与通常的科学观念作对。我们多么不想承认这种说法呀！然而却存在着无可辩驳的统计证据，使人们至少对心灵感应不得不信。想要重新调整人们已有的观念从而接受这些新事物谈何容易。我们一旦接受了这些观念，离相信妖魔鬼怪也为期不远了。跨出的第一步就是相信我们的身体除了简单地按照已知的物理学规律运动外，同时还会受一些未知但相近的规律影响。

这个论点在我看来十分有说服力。我们可以这样回答，许多科学理论尽管同超感知觉有冲突，但在实际世界中却是可行的；事实上，它们彼此共存并不矛盾。这是一种冷漠的安慰，人们唯恐思维与超感知觉之间真的有着特殊的联系。

基于超感知觉，有一个更具体的论证："在模仿游戏中，让一个善于接受心灵感应的人和一台数字计算机作为证人。提问者可以这样问，例如说'我右手中的那张牌是哪个花色？'有心灵感应或超视觉的人在 400 张牌中可以猜对 130 张，而机器只能随机猜中 104 张左右。因此，提问者就能够正确判断了。"有意思的是，还存在另一种可能。假使这台数字计算机有一个随机数字生成程序，那么，它很自然就能利用这个程序作答。但是，这个随机数字生成程序又将被提问者意念运动的力量所支配。在意念运动的作用下，很有可能计算机猜对的次数比概率计算的还要高。结果就是提问者仍旧无法正确辨识。再换个角度思考，提问者不也可能有超视觉吗？甚至不用提问就能辨识机器与人。有了超感知觉，一切皆有可能。

如果心灵感应被承认的话，那我们的游戏设定就得更严格一些。现在这个情景就好比提问者在屋子里自言自语，而隔壁的被问者正竖着耳朵贴墙偷听。要是能让被问者待在"防心灵感应室"里头，那么游戏的要求就都满足了。

7. 学习机器

读者可能会认为，我缺乏令人信服的实证论据，不足以支持自己的观点。否则的话，我何必大费周章地逐一反驳对立的观点呢？其实并非如此，现在才是我要拿出证据的时候。

先回到拉芙蕾丝夫人的观点，她认为机器只能按我们的指示做事。有人可能会说，可以给机器“注入”一个想法，机器会在一定程度上做出反应，最后回归静态，就像一个被锤子敲击的钢琴弦一样。或者说，就像一个小于临界尺寸的原子反应堆：输入的概念就像从原子反应堆外部进入的中子。这些中子会引起一些干扰，最后逐渐消失。但是，如果原子反应堆的规模持续扩大，中子引起的干扰也很可能会持续增加，直到原子反应堆解体。人类的思维中是否存在对应的现象？机器呢？这样的现象在人类的思维中似乎是存在的，而且绝大多数都处于“亚临界”状态，类似于处于亚临界尺寸的原子反应堆。一个概念进入这样的思维中，回馈的概念平均可能连一个都不到。只有一小部分处于超临界状态，进入其中的概念将会产生二级，三级以及越来越多的概念，最终形成一套完整的“理论”。动物的头脑显然是处于亚临界状态的。由于这种相似性，我们不得不问：“一个机器能不能被构造成超临界状态？”

“洋葱皮”的比喻也很有用。研究思维或大脑的功能时，我们发现一些操作完全可以从纯机械的角度加以解释。这并不是真正的思维，而只是一层表皮。为了发掘真正的思维，我们必须把它像洋葱皮一样剥除。但在这时，我们发现丢弃的洋葱皮里，包裹着的仍然是需要剥除的机械思维，以此类推。用这样的方式，我们是否能够发掘“真正的”思维呢？或者最终发现，最后一层洋葱皮里空空如也？如果是第二种情况，那么整个思维

都是机械的。（但它不一定是离散状态机，我们已经讨论过这一点。）

上面的两段文字，与其说它们是有力的证明，倒不如称之为“为了滋生信仰的诵读”。

在第六节的开头部分，我们提出了一个观点，唯一真正能符合这个观点要求的论据，看来只能等到本世纪末再进行实验了。实验的同时，我们还能够证明些什么呢？要让这个实验成功，我们现在应该准备些什么呢？

正如我所解释的那样，程序的编写是关键。工程设计上的进步也很重要，而且看来它们完全可以满足要求。我们估计，大脑的存储容量在 10^{10} 位到 10^{15} 位之间。我个人比较倾向于保守的估计，因为我认为只有很小一部分存储容量被用来进行高级的思维。其余的大部分可能用来保存视觉图象。我不认为模仿游戏需要的存储容量会超过 10^9 位，这至少足够对阵一位盲人了。[①]即使仅立足于目前的技术，10^7 位的存储容量已经可行了。也许，根本就不需要提高机器的执行速度。有一些模拟神经细胞的现代机器的速度极快，比神经细胞快 1000 倍。这样的“安全边际”可以补偿由各种因素导致的速度损失。接下来的主要问题就是找到给机器编程的方法，让它们能够完成游戏。现在我一天大概能编 1000 字的程序，那如果 60 个码农能在未来 50 年内稳定、高效、不浪费地工作，就有可能完成这项任务。越快越好。

在模仿一个成人思维的时候，我们必须考虑它是怎样达到当前状态的。我们会发现有以下三个部分：

(a) 思维的初始状态，也就是出生时的状态

(b) 它所接受的教育

① 图灵原注：第十一版英国大百科全书的容量为 2×10^9 位。

(c) 它的教育之外的经历

与其尝试设计一个程序模仿成人的思维，为何不试试模仿儿童的思维呢？如果它接受合适的教育，它就可能成长为一个成人的大脑。一个儿童的大脑就像一本刚从文具店买来的笔记本，由简单的机制和许多空白页组成。（机制和在纸上写字在我们看来几乎是一样的。） 我们希望儿童脑中只有极少的机制，这样编程要容易些。我们猜教育机器的工作量和教育一个人类儿童应该基本是一样的。

这样一来，问题就被分解为了两个部分：设计一个儿童程序以及教育它。这两者是联系紧密的。要一下子就找到一个合适的模拟儿童的机器是不可能的。我们必须通过实验教学来研究机器的学习效果，接着再试验另一个程序并判断孰好孰坏。通过定义识别，看得出来这个过程和人类进化有着明显的相关性。

儿童模拟机的结构 = 遗传物质

儿童模拟机的变化 = 变异

自然选择 = 实验者的决定

然而有人可能期待，这个过程会比人类自然进化快上许多。适者生存是度量优势的一个方法，但很缓慢。实验者通过施加自己的智能，有可能加快这一过程。同样重要的是，实验者并不会局限于随机的变异。只要能够追踪到缺陷产生的原因，实验者或许就可以对症下药，通过另一种变异去改良这种缺陷。

教育机器和教育正常儿童的过程不会完全相同。例如，机器没有腿，因此不能要求它去给煤斗添煤。况且机器很可能也没有眼睛。不管聪明的工程师采取何种方法克服这些缺陷，只要是机器被送进人类的学校，肯定会被其他学生加以嘲笑。机器必须得到专门的指导才行。我们不需要把注

意力过多地放在腿、眼等器官上。海伦·凯勒小姐的例子已经表明，只要老师和学生能够以某种方式进行双向的交流，教育就能进行。

通常的教学过程是惩罚和奖励相结合的。我们可以利用这些原则，建造或编写简单的儿童机器。如果一个行动刚刚执行就受到惩罚，机器要能不再重复这个行动；而在接受奖励的时候，要能提高这个行动发生的概率。这些定义并不能预设机器的情绪。我对一台儿童机器进行了一些实验，而且成功地教会了它一些东西。但是教育方法不太正规，所以还称不上成功。

惩罚和奖励最好能作为机器教育过程的一部分。粗略地说，如果老师没有其他和学生交流的方式，学生接收到的信息，不会超过奖励和惩罚时教授信息的总和。一个孩子要学会重复“Casabianca”这个词，一定会是个痛苦的经历。如果教学上只能通过“提二十个问题”的方法来传授这个词的话，每一个“No”都将是一个沉重的打击。因此应该寻找其他“非情绪化”的交流渠道。如果能够找到的话，那么就可以通过惩罚和奖励的方式，让机器学会服从某种语言发出的命令，例如符号语言。这些命令则通过“非情绪化”的渠道传输。这种语言的使用将会大大降低必需的惩罚和奖励的次数。

什么样的复杂程度更适合儿童机器呢？不同的人会有不同的看法。有人主张在遵守普遍原则的前提下，越简单越好。另一种观点是嵌入一个完整的逻辑推理系统。在第二种情况下，大部分的存储空间将被用来存储定义和命题。这些命题可能具有各种各样的状态，例如，确定的事实、猜想、数学上证明的定理、权威给出的判断、具有命题逻辑形式却没有确定值的表达式，等等。一些命题可以被称为“命令”。机器的构造要使它能够在命令得到确定的时候，立即自动采取合适的行动。举个例子，如果老师对机器说“现在做家庭作业”，那么“老师说‘现在做家庭作业’”将成

为事实。另一个事实可能是“老师说的一切都是对的”。把这俩事实结合在一起，将会使“现在做家庭作业”这个命令被纳入确定的事实之列。根据机器的建造规则，这意味着它将立即开始写家庭作业，而且结果会令人非常满意。机器的推理过程其实并不一定非得让最严格的逻辑学家满意。例如，机器中可能不存在类型体系。即使如此，类型悖论出现的几率，也不会高过我们从未设栏杆的悬崖上摔下的概率。合理的命令（在系统内部表达，并不是系统规则的一部分）之间常常具有相似的效果，比如“不要使用一个类（class），除非它是老师提到某个类的一个子类（subclass）”和“不许接近边缘”这两个命令就是如此。

要让一台没有四肢的机器能够服从指令，这必然是智力层面的事，就像上面举的做家庭作业的例子。在这些指令中，重要的是规定应用于逻辑系统的规则以什么样的顺序执行。因为在这个系统的每个阶段，都会有大量不同的步骤可供选择。在遵守逻辑系统规则的情况下，选择任意一个都是允许的。这些选择所体现的将是智慧和愚昧推理者的差距，而不是正确或谬误的不同。指令的命题可能会是这样：“当提到苏格拉底的时候，使用芭芭拉三段论”（Barbara syllogism）[①]或者“如果有一个方法被证明比其他方法快捷，不要用慢的方法”。这些陈述有一部分可能“来自权威”，其他的有可能来自机器本身，例如通过科学归纳。

一些读者可能会觉得会学习的机器这个想法有些矛盾。怎样可能改变机器运行的规则呢？这些规则决定机器会做出的所有反应，不管它的过去曾经历过什么变化。因此，所有的规则应该是不随时间改变的。确实是这样。对这个悖论的解释是，在学习过程中改变的规则都是那些不太自命

① 译注：所谓“芭芭拉三段论”是指“所有人都必死，所有希腊人是人，故所有希腊人必死”。在逻辑教科书里，常说“所有人必死，苏格拉底是人，故苏格拉底必死”。

不凡且短视的规则。读者可以拿美国宪法做比较。

学习机有一个重要的特点，它的老师通常不关心机器内部发生了什么变化，尽管在一定程度上能预测学生的行为。在设计精良的儿童计算机（或程序）的后期教育上更应该如此。这和机器计算时使用的常规操作程序形成了鲜明的对比，计算时的目的是对机器在任意时刻的状态有清晰的了解。这个目标很难达到。“机器只能按我们的指令行动”①的观点在这里就说不过去了。我们输入机器的大部分程序将导致机器执行一些我们无法完全理解的事，或者在我们看来完全随机的事。智能行为和计算中完全服从命令的行为可能有些不同，但这种区别不大，不至于产生随机行为或是无意义的循环。另外还有一点非常重要，当我们的机器通过教与学的方式“备战”模仿游戏时，“人类不靠谱”的特点很可能被自然地忽略，也就是不再需要专门的“辅导”。（读者应该将此与本文第 24、25 页上描述的观点调和一下。）学到的方法很难产生百分之百确定的结果；如果可以的话，那么这些方法就不会被遗忘了。

在一个学习机器中加入随机元素也许是个明智的做法。随机元素可以帮助我们寻找问题的答案。例如我们想找一个介于 50 和 200 之间的数，这个数的值等于它的各个数字的和的二次方。我们可以从 51 开始，再到 52，一直试验下去，直到找到满足条件的数。另一个方法是随机抽选，直到找到满足条件的数。这个方法的优点是不需要记录已经验算过的数值，但缺点是可能会出现重复计算。当题目有多解的时候，这个缺陷就被弱化了。系统化方法有一个缺点，那就是在很大一个区间内很有可能不存在解，但我们却仍然需要一一验证。现在的学习过程可以看成在寻找一个能够满

① 图灵原注：与拉芙蕾丝夫人的论断比较（第 450 页，（译注：第六节开始）），她的原话并没有“只能”。

足老师的要求（或其他的标准）的行为规则。由于可能存在大量的解，随机的方法可能更优于系统的方法。在类似进化的过程中，也运用到了随机的方法，系统化的方法是行不通的。那么我们得思考，应该如何记录已经试验过的不同的基因组合以避免重复的无用功呢？

最终我们可能希望看到，机器能够纯粹在智力领域和人类一较高下。但是从什么领域开始呢？这也是个难题。许多人建议选择抽象的项目最好，例如国际象棋。也有人认为要给机器提供能买得到的最好的传感器，再培养它理解和学说英语。这个过程和教一个正常的孩子是一样的。我们会告诉它们，这是什么，那是什么，等等。我仍然不知道正确的答案是什么，但是我想这两种方法都应该试试。

初见前路近可至，细思百事竟待忙。（We can only see a short distance ahead, but we can see plenty there that needs to be done.）

参考文献

Samuel Butler, Erevhon, London, 1865. Chapters 23, 24, 25, *The Book of the Machines*.

Alonzo Church, “An Unsolvable Problem of Elementary Number Theory”, *American J. of Math.*, 58 (1936), 345-363.

K. Gödel, “Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I”, *Monatshefte für Mathematik und Physik., (1931) 173–198.*

D. R. Hartree, *Calculating Instrutments and Machines*, New York, 1949.

S. C. Kleene, “ General Recursive Functions of Natural Numbers ”, *American J. of Math.*, 57 (1935), 153-173 and 219-244.

G. Jefferson, “ The Mind of Mechanical Man ”. Lister Oration for 1949. *British Medical Journal,* vol. i (1949), 1105-1121.

Countess of Lovelace, “Translator’s notes to an artlcle on Babbage's Analytical Engine”, *Scientific Memoirs* (ed. by R. Taylor), vol. 3 (1842), 691-731.

Bertrand Russell, *History of Western Philosophy,* London, 1940.

A. M. Turing, " On Computable Numbers, with an Application to the Entscheidungsproblem ", *Proc. London Math. Soc.* (2), 42 (1937), 230-265.

于维多利亚曼彻斯特大学[①]（*Victoria University of Manchester*）.

① 译注：就是现在的曼彻斯特大学。

参考文献

Aaronson, Scott (2013), *Quantum Computing since Democritus.*

Aaronson, Scott (2013), Why Philosophers Should Care about Computational Complexity, *Computability: Turing, Godel, Church and Beyond*, ed. Jack Copeland, Carl Posy and Oron Shagir, MIT Press.

Aaronson, Scott, Who Can Name the Bigger Number, http://www.scottaaronson.com/writings/bignumbers.html.

Allemang, Dean and Jim Handler (2011), *Semantic Web for Working Ontologist*, 2nd ed.（语义万维网——工程实践指南, 高等教育出版社, 2014）.

Anderson, James and Edward Rosenfeld (1998), *Talking Nets: An Oral History of Neural Networks*, Bradford Books.

Arbib, Michael (1987), *Brains, Machines and Mathematics*, Springer, 1st ed 1964, 2nd ed（大脑, 机器和数学, 商务印书馆, 1982）

Arora, Sanjeev and Boaz Barak (2009), *Computational Complexity: A Modern Approach*, Cambridge University Press.

Ballantyne, M., R. S. Boyer and L. Hines (1996), Woody Bledsoe: His Life and Legacy, *AI Magazine*, Vol. 17, No. 1, pp. 7-20, Springer.

Batterson, Steve (2000), *Stephen Smale: The Mathematician Who Broke the Dimension Barrier*, American Mathematical Society.

Beeson, Michael J. (1995), *Computerizing Mathematics: Logic and Computation*, in Rolf Herken ed.

Bellman, R. E. (1957), *Dynamic Programming*, Princeton University Press.

Benenti, Giuliano, Giulio Casati and Giuliano Strini (2004), *Principles of Quantum Computation and Information*, Vol. 1.（量子计算与量子信息原理, 第一卷：基本概念, 科学出版社, 2011）

Berners-Lee, Tim, Jim Hendler and O. Lassila (2001), The Semantic Web. *Scientific American,*

284 (5):34–43.

Berwick, Robert and Noam Chomsky (2016), *Why Only Us*, MIT Press.

Bibel, W. (1987), *Automated Theorem Proving*, 2nd ed, Verlag.

Bibel, W. (2007), Early History and Perspectives of Automated Deduction, J. Hertzberg, M. Beetz and R. Englert (eds) *Lectures Notes on Artificial Intelligence*, Springer-Verlag.

Birkhoff G. and S. MacLane (1961), *A Survey of Modern Algebra*.

Blake, D. V. and A. M. Uttley (1959), (ed.) *Proceedings of the Symposium on Mechanisation of Thought Process*, H. M. Stationery Office.

Blum, L., F. Cucker, M. Shub and Steve Smale (1998), *Complexity and Real Computation*, Springer-Verlag.

Boden, Margaret (2008), *Mind as Machine: A History of Cognitive Science*, Oxford University Press.

Boden, Margaret (2016), *AI: Its Nature and Future*.（人工智能的本质与未来, 中国人民大学出版社, 2017）

Bolhuis, Johan, Ian Tattersall, Noam Chomsky and Robert C. Berwick (2014), How Could Language Have Evolved? *PLoS Biology*, 12(8): e1001934. doi: 10.1371/journal.pbio.1001934.

Bonacina, Maria Paola and Mark E. Stickel (2013), Automated Reasoning and Mathematics, *Essays in Honor of William McCune,* Springer.

Boyer, R. S. and J. S. Moore (1977), A Fast String Searching Algorithm, *Communications of ACM*, Vol. 20, No. 10, pp. 762–772.

Boyer, R. S. and J. S. Moore (1979), *A Computational Logic Handbook*, Academic Press.

Brooks, Rodney (2001), The relationship between matter and life, *Nature*, Vol. 409, 18 Jan.

Brooks, Rodney (2013), *The Case for Embodied Intelligence*, in Cooper and Leeuwen.

Brown, Peter F., John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer and Paul S. Roossin (1988), A Statistical Approach to Language Translation, *Proceesings of 12th Conference of Computational Linguistics.*

Brown, Peter F., John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer and Paul S. Roossin (1990), A STATISTICAL APPROACH TO MACHINE TRANSLATION, *Computational Linguistics*, Vol. 16, No. 2.

Buchanan, Bruce G. (1991), Oral history interview with Bruce G. Buchanan, Charles Babbage Institute, http://conservancy.umn.edu/handle/11299/107165.

Buchanan, Bruce G. (2006), A (Very) Brief History of Artificial Intelligence, *AI Magazine*, Vol. 26, No. 4.

Buchanan, Bruce G. and Edward H. Shortliffe (1984), (ed.) *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison Wesley.

Burks, Arthur W. and Hao Wang (1957), The Logic of Automata, parts I and II, *Journal of the*

Association for Computing Machinery.

Burris, Stanley N. (1997), *Logic for Mathematics and Computer Science*, Prentice Hall.

Calvin, William (1996), *How Brains Think: Evolving Intelligence, Then and Now.*

Calvin, William (2004), *A Brief History of the Mind: From Apes to Intellect and Beyond..*

Chaitin, Gregory (2010), To a Mathematical Theory of Evolution and Biological Creativity, *CDMTCS Research Report Series*, https://www.cs.auckland.ac.nz/research/groups/CDMTCS/researchreports/391greg.pdf.

Chaitin, Gregory (2012), *Proving Darwin: Making Biology Mathematical,* Vintage Books,(证明达尔文, 人民邮电出版社, 2015)

Chang, C. L. and Richard Char-tung Lee (1973), *Symbolic Logic and Mechanical Theorem Proving*, Academic Press.

Chiou, Stefanie, Craig Music Kara Sprague and Rebekah Wahba (2001), A Marriage of Convenience: THE FOUNDING OF THE MIT ARTIFICIAL INTELLIGENCE LABORATORY, http://web.mit.edu/6.933/www/Fall2001/AILab.pdf.

Chomsky, Noam (1953), Systems of Syntactic Analysis, *Journal of Symbolic Logic*, 18, No. 3.

Chomsky, Noam (1956), Three Models for the Description of Language, *Proceedings of Symposium on Information Theory*.

Chomsky, Noam (1957), *Syntactic Structures*.(句法结构, 中国社会科学出版社, 1979)

Chomsky, Noam (1975), *The Logical Structure of Linguistic Theory.*

Chomsky, Noam (1986), *Knowledge of Language: Its Nature, Origins, and Use.*

Chomsky, Noam (2011), *The Science of Language, Interviews with James McGilvray*, Cambridge University Press.(语言的科学, 商务印书馆, 2015)

Chou, shang-Ching Chou (1987), *Mechanical Geometry Theorem Proving*, Springer.

Church, Alonzo (1941), *Calculi of Lambda conversion*, Princeton University Press.

Cohen, Paul and Edward Feigenbaum (1983), *Handbook of Artificial Intelligence*, Vol. 3.

Colby, K. (1975), *Artificial Paranoia*, Pergamon.

Cooper, S. Barry and Andrew Hodges (2016), *The Once and Future Turing: Computing the World*, Cambridge University Press.

Cooper, S. Barry and J. V. Leeuwen (2013), *Alan Turing: His Work and Impact*, Elsevier.

Copeland, J. Carl Posy and Oron Shagrir (2013), *Computability: Turing, Godel, Church, and Beyond*, MIT Press.

Copeland, Jack (1993), *Artificial Intelligence: A Philosophical Introduction*, Blackwell.

Copeland, Jack (2012), *Turing: Pioneer of the Information Age*. Oxford University Press.

Crick, Francis (1994), *The Astonishing Hypothesis: The Scientific Search for the Soul*, Simon and Schuster.

Critchley, Simon (2009), *The Book of Dead Philosophers*.

Darden, Lindley (1987), Viewing the history of scicncc as compiled hindsight. *Al Magazine*, Summer.

Dauben, Joseph (2002), Book Review: The Universal History of Numbers and The Universal History of Computing (part 1), (part2), *Notices of the AMS.* Vol. 49, Jan. And Feb.

Davis, M. (2011), Hao Wang's contributions to mechanized deduction and to the Entscheidungs problem, *Hao Wang: Logician and Philosopher*, (ed.) Charles Parsons and Montgomery Link, Individual Authors and College Publications.

Davis, M. (2008), Interview with Martin Davis, *Notices of the AMS*, May, 2008.

Davis, M. (2001), The Early History of Automated Deduction, *Handbook of Automated Reasoning*.

Davis, M. (1957), A computer program for Presburger's algorithm. In *Summaries of talks presented at the Summer Institute for Symbolic Logic*, Princeton, pp. 215–233, Institute for Defense Analysis, in Siekmann and Wrightson, pp. 41–48.

de Brujin, N. G. (1980), A survey of the project AUTOMATH, in *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, (eds) J. P. Seldin and J. R. Hindley, Academic Press.

Dennett, D. (1997), *Kinds of Minds: Toward an Understanding Of Consciousness*, Basic Books.

Deutsch, David (1998), *The Fabric of Reality: The Science of Parallel Universes and Its Implications*, Penguin.（真实世界的脉络, 人民邮电出版社, 2016）

Deutsch, David (2011), *The Beginning of Infinity: Explanations that Transform the World.*（无穷的开始：世界进步的本原, 人民邮电出版社, 2016）

Djerassi, Carl (1992), *The Pill. Pygmy Chimps, and Degas' Horse: The Autobiography of Carl Djerassi*, Basic Books.

Dormehl, Luke (2016), *Thinking Machines: The Quest for Artificial Intelligence--and Where It's Taking Us Next*.（人工智能：改变世界, 重建未来, 中信出版社, 2017）

Dreyfus, Hubert (1965), *Alchemy and AI*, RAND Corporation.

Dreyfus, Hubert (1972), *What Computers Can't Do: The Limits of Artificial Intelligence,* MIT Press.（计算机不能做什么, 三联书店）

Dreyfus, Hubert (1992), *What Computers Still Can't Do: The Critique of Artificial Reason,* MIT Press.

Dreyfus, Hubert (2001), *On the Internet*, Revised Second Edition, Routledge.

Dreyfus, Hubert (2008), Why Heideggerian AI Failed and how Fixing it would Require making it more, in Husbands et al.

Dreyfus, Hubert and Stuart Dreyfus (1988), Making a Mind Versus Modeling the Brain: Artificial Intelligence Back at a Branchpoint, in Graubard.

Dunham, B., Fridshal, R., Sward, G. L. (1960), A non-heuristic program for proving elementary logical theorems. In *First International Conference on Information Processing*, Paris, pp. 282–285. Unesco House, in Siekmann and Wrightson, Vol. 1, pp. 93–98.

Dyson, Freeman (2008), *The Scientist as Rebel*, New York Review Books.（反叛的科学家, 浙江大学出版社, 2013）

Feferman, Solomon (2005), Tarski's influence on computer science, Invited Talk, *LICS*, Chicago.

Feigenbaum, E. and J. Feldman (1963), *Computers and Thought*, McGraw Hill.

Feynman, Richard (1982), Simulating Physics with Computers, *International Journal of Theoretical Physics,* Vol. 21, Nos 6/7.

Feynman, Richard (2000), *Feynman Lectures on Computation*, ed. Anthony Hey and Robin Allen, revised edition.

Fisher, Amy Weaver and J. L. MC Kenney (1993), The Development of the ERMA Banking System: Lessons from History, *IEEE Annals of the Hirrory of Computing*. val.15, No.1.

Fisher, Ronald (1930), *The Genetical Theory of Natural Selection.*

Fitting, M. (1995), *First-Order Logic and Automated Theorem Proving*, 2nd ed., Springer.

Friedberg, R. M. (1958), A Learning Machine: Part 1, *IBM Journal of Research and Development*, Vol. 2, January, pp. 2-13.

Frost, Richard (1986), *Introduction to Knowledge Base Systems*, Collins Professional and Technical Books, London.

Fuchi, Kazuhiro and Koichi Furukawa (1987), The Role of Logic Programming in the Fifth Generation Computer Project, New Generation Computing, March 1987, Vol. 5, Issue 1, pp. 3-28.

Furukawa, Koichi (1987), Fifth generation computer project: Current research activity and future plans, TAPSOFT'87, LNCS, Vol. 250, Springer.

Gallier, J. H. (2003), *Logic for Computer Science: Foundations of Automatic Theorem Proving*, 2nd ed., Dover.

Gardner, Howard (1987), *The Mind's New Science: A History of the Cognitive Revolution*, Basic Books.

Garnelo, Marta, Kai Arulkumaran and Murray Shanahan (2016), *Towards deep symbolic reinforcement learning.* arXiv:1609.05518.

Gessler, Nicholas (2013), The Computerman, The Cryptographer and The Physicist, in Cooper and Leeuwen.

Gibson, David and Everett Rogers (1994), *R & D Collaboration on Trial: The Microelectronics and Computer Technology Corporation*, Harvard University Press.

Gilmore, P. C. (1960), A proof method for quantification theory: Its justification and realization, *IBM Journal of Research Development*, Vol. 4, pp. 28–36.

Goldberg, David (1989), Genetic Algorithms in Search, Optimization and Machine Learning, *Reading*, MA: Addison-Wesley Professional.

Goldreich, O. (2004), On Quantum Computing, http://www.wisdom.weizmann.ac.il/~oded/on-qc.html.

Goto-Jones, Christopher (2007), *Modern Japan: A Very Short Introduction*, Oxford University Press.（现代日本, 译林出版社, 2014）

Graf, P. (1996), *Term Indexing*, Springer-Verlag.

Graubard, Stephen (1988), (ed.) *The Artificial Intelligence Debate: False Starts, Real Foundations*, MIT Press.

Graves, Alex, Greg Wayne and Ivo Danihelka (2014), Neural Turing Machine, arXiv:1410.5401v2 [cs.NE] 10 Dec 2014.

Guha, Ramanathan (2013), Light at the End of Tunnel, http://videolectures.net/iswc2013_guha_tunnel/.

Hadamard, Jacques (1954), An essay on the psychology of invention in the mathematical field, New York: Dover Publications.（数学领域中的发明心理学, 江苏教育出版社, 1988）

Halmos, Paul (1988), *I Want to Be a Mathematician: An Automathography*, Mathematical Association of America.

Harari, Yural Noal (2016), *Homo Deus: A Brief History of Tomorrow*.（未来简史, 中信出版社, 2017）

Hassabis, Demis, Dharshan Kumaran, Christopher Summerfield and Matthew Botvinick (2017), Neuroscience-Inspired Artificial Intelligence, *Neuron*.

Heath, David and Derek Allum (1997), The Historical Development of Computer Chess and its Impact on Artificial Intelligence, *AAAI Workshop: Deep Blue Versus Kasparov: The Significance for Artificial Intelligence*.

Hebb, Donald Olding (1949), *The Organization of Behavior: A Neuropsychological Theory*, Wiley.

Hecht-Nielsen, Robert (1989), Theory of the backpropagation neural network, *International Joint Conference on Neural Networks*, pp. 593–605.

Henderson, Harry (2007), *Artificial Intelligence: Mirrors for the Mind*, Chelsea House Pulications.

Herken, Rolf (1994), (ed.) *The Universal Turing Machine: A Half Century Survey*, Springer-Verlag.

Hewitt, Carl (2009), Middle History of Logic Programming, https://arxiv.org/pdf/0904.3036v4.pdf.

Hinton, Geoffrey E. and P. R. Salakhutdinov (2006), Reducing the Dimensionality of Data with Neural Networks, *Science*, 313, 5786, pp. 504-507.

Hinton, Geoffrey E., Simon Osindero and Yee-Whye Teh (2006), A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7) pp 1527–1554.

Hodges, A. (1983), *Alan Turing: The Enigma*, Simon & Schuster.

Hodges, Wilfrid (2007), Necessity in mathematics, http://wilfridhodges.co.uk/semantics06.pdf.

Holland, John (1959), *Cyclesin Logical Nets*, PhD Thesis, University of Michigan, https://deepblue.lib.umich.edu/bitstream/handle/2027.42/32404/0000479.pdf.

Holland, John (1975), *Adaptation in Natural and Artificial Systems: An Introductory Analysis*

with Applications to Biology, Control and Artificial Intelligence, 1st edition, University of Michigan Press.

Holland, John (1999), *Emergence: From Chaos to Order*.

Holland, John (2008), An Interview with John Holland, in Husbands, Philip, Owen Holland and Michael Wheeler.

Holland, John (2014), *Complexity: A Very Short Introduction*, Oxford University Press.

Hong, Hoon, Deepak Kapur and Peter Paule (2006), Bruno Buchberger—A life devoted to symbolic computation, *Journal of Symbolic Computation*, Vol. 41, pp. 255-258.

Hong, Jia-wei (1986), *Computation: Computability, Similarity and Duality*, Pitman.

Hopfield, J. J. and Tank, D. W (1985), "Neural" computation of decisions in optimization problems. *Biological Cybernetics,* 55, pp. 141-146.

Hopfield, John (1982), Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences*, 79(8) pp. 2554–2558.

Horgan, John (1990), *The End of Science: Facing The Limits Of Knowledge In The Twilight Of The Scientific Age.*

Horgan, John (1993), The Death of Proof, *Scientific American*, Oct., 1993.

Hsu, Feng-hsiung (2004), *Behind Deep Blue: Building the Computer that Defeated the World Chess Champion*, Princeton University Press.（"深蓝"揭秘：追寻人工智能圣杯之旅, 上海科技教育出版社, 2005）

Huang, Xuedong, James Baker and Raj Reddy (2014), A Historical Perspective of Speech Recognition, *CACM*, Vol. 57 No. 1, pp. 94-103.

Husbands, Philip, Owen Holland and Michael Wheeler (2008), (eds) *The Mechanical Mind in History*, MIT Press.

Hutchins, John (2005), The history of machine translation in a nutshell, http://www.hutchinsweb.me.uk/Nutshell-2005.pdf.

IBM J. Res. Develop., 11.

ICOT (1988), (ed.) *Fifth Generation Computer Systems*, 1988 FGCS'88, Tokyo.

Ifrah, Georges (1994), *From One to Zero: A Universal History of Numbers.*

Ifrah, Georges (2001), *The Universal History of Computing: From the Abacus to the Quantum Computer*.

Jacobsen, Annie (2015), The Pentagon's Brain, *An Uncensored History of DARPA,* America's Top-Secret Military Research Agency.

Jain, Sanjay, Daniel N. Osherson, James S. Royer and Arun Sharma (1999), *Systems That Learn: An Introduction to Learning Theory (Learning, Development, and Conceptual Change)*, 2nd Edition, MIT Press.

Johnson, Paul (1988), *Intellectuals: From Marx and Tolstoy to Sartre and Chomsky,* Weidenfeld & Nicolson.

Judd, Stephen (1990), *Neural Network Design and the Complexity of Learning*, MIT Press.

Kakas, Antonis C. and Fariba Sadri (2002), (ed.), Computational Logic: Logic Programming and Beyond, *Essays in Honour of Robert Kowalski*, Part 1, Springer.

Kalman, J. A. (2001), *Automated Reasoning with Otter*, Rinton Press.

Kasparov, Garry (2016), *Deep Thinking: Where Machine Intelligence Ends and Human Creativity Begins.*

Katz, Yarden (2012), Noam Chomsky on Where Artificial Intelligence Went Wrong, *Atlantic,* Nov. 1.

Kemeny, John (1955), Man Viewed as a Machine, *Scientific American*, Vol. 192, No. 4.

Knuth, Donald and Bendix, P. B. (1970), Simple Word Problems in Universal Algebras. In *Computational Problems in Abstract Algebra*. J. Leech, 263-297. Oxford, U. K.: Pergamon.

Knuth, Donald (2001), *Things a Computer Scientist Rarely Talks About*, Center for the Study of Language and Information.

Kolata, Gina, *New York Times*, http://www.nytimes.com/library/cyber/week/1210math.html, 1996.

Korf, R. (1997), Does Deep-Blue use AI? *AAAI Workshop: Deep Blue Versus Kasparov: The Significance for Artificial Intelligence*.

Kowalski, Robert (1986), The Limitations of Logic. *Proceedings of the ACM Annual Conference on Computer Science*.

Kowalski, Robert (1988), The Early Years of Logic Programming, *CACM*, January 1988.

Koza, J. R. (1990), *Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems*, Stanford University Computer Science Department technical report STAN-CS-90-1314. http://www.genetic-programming.com/jkpdf/tr1314.pdf.

Koza, J. R. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press.

Koza, J. R. (1994), *Genetic Programming II: Automatic Discovery of Reusable Programs*, MIT Press.

Kruozumi, Takashi (1992), Overview of the 10 years of the FGCS project. *Proceedings of the international conference on fifth generation computer systems*.

Kubat, Miroslav (2015), *Introduction to Machine Learning*, Springer.（机器学习导论, 机械工业出版社, 2016）

Lassez, Jean-Louis and Gordon Plotkin (1991), (ed.) *Computational Logic: Essays in Honor of Alan Robinson*, MIT Press.

LeCun, Yann, Yoshua Bengio and Geoffrey Hinton (2015), Deep learning, *NATURE*, Vol. 521, 28 May 2015, http://www.cs.toronto.edu/~hinton/absps/Nature Deep Review.pdf.

Lederberg, Joshua (1987), How DENDRAL was conceived and born. Rockefeller University, *ACM Symposium on the History of Medical Informatics.*

Lenat, Douglas and R. V. Guha (1989), *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*.

Lindsay, Robert K., Bruce G. Buchanan, Edward A. Feigenbaum and Joshua Lederberg (1993), DENDRAL: a case study of the first expert system for scientific hypothesis formation, *Aritificial Intelligence*, Vol. 61, pp. 209.

Livnat, Adi and Christos Papadimitriou (2016), Sex as an Algorithm: The Theory of Evolution Under the Lens of Computation, *Communications of the ACM*, Vol. 59 No. 11, Pages 84-93.

Loos, Sarah, Geoffrey Irving, Christian Szegedy and Cezary Kaliszyk (2017), Deep Network Guided Proof Search, arXiv:1701.06972v1 [cs.AI] 24 Jan 2017.

Loveland, D.W. (1978), *Automated Theorem Proving: A Logical Basis,* North Holland.

Lungarella, M., F. Iida, J. Bongard and R. Pfeifer (2007), *50 Years of Artificial Intelligence: Essays Dedicated to the 50th Anniversary of Artificial Intelligence*, Springer.

Machlup, Fritz and U. Mansfield (1983), (ed.) *The Study of Information: Interdisciplinary Messages*, John Wiley.

MacRae, Norman (1997), *John Von Neumann: The Scientific Genius Who Pioneered the Modern Computer, Game Theory, Nuclear Deterrence, and Much More*, 2nd ed., AMS.

Marr, David (1982), *Vision: A Computational Approach,* Freeman & Co.

McCarthy, John (1979), History of LISP, http://jmc.stanford.edu/articles/lisp/lisp.pdf.

McCarthy, John (1988), Mathematical Logic in Artificial Intelligence, in *The Artifical Intelligence Debate: False Starts, Real Foundations*, ed Stephen R. Grubard, MIT Press.

McCarthy, John (1989), Oral History Interview with John McCarthy, Charles Babbage Institute, http://conservancy.umn.edu/handle/11299/107476.

McCarthy, John and Hayes, P. J. (1969), Some philosophical problems from the standpoint of artificial intelligence at the Wayback Machine, In Meltzer, B. and Michie, D., eds., *Machine Intelligence* 4.

McCarthy, John, M. Minsky, N. Rochester and C. Shannon (1955), A PROPOSAL FOR THE DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE, August 31, 1955.

McCarthy, John (2005), Science Lives, interview at Simons Foundation, https://www.simonsfoundation.org/science_lives_video/john-mccarthy/.

McCorduck, Pamela (1979), *Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence*, Freeman and Company.

McCorduck, Pamela and Edward Feigenbaum (1984), *The Fifth Generation: Artificial Intelligence & Japan's Computer Challenge to the World*, 2nd ed, Addison Wesley.

McCulloch, Warren S. and Walter Pitts (1943), A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

Milburn, Gerald (1999), *The Feynman Processor: Quantum Entanglement and the Computing Revolution*, Basic Books.

Minsky, Marvin (1974), A Framework for Representing Knowledge, MIT AI Lab Memo 306, http://web.media.mit.edu/~minsky/papers/Frame/frames.html.

Minsky, Marvin (1989), Oral History Interview with Marvin Minsky, Charles Babbage Institute, http://conservancy.umn.edu/handle/11299/107503.

Minsky, Marvin L. and Seymour A. Papert (1969), *Perceptrons: an introduction to computational geometry*, MIT Press.

Mirhoseini, Azalia, Hieu Pham, Quoc V. Le, Benoit Steiner, Rasmus Larsen, Yuefeng Zhou, Naveen Kumar, Mohammad Norouzi, Samy Bengio and Jeff Dean (2017), Device Placement Optimization with Reinforcement Learning, arXiv:1706.04972 [cs. LG].

MIT, MIT 150 Symposia on Brains, Minds and Machines, http://mit150.mit.edu/symposia/brains-minds-machines, 2011.

Mitchell, Melanie (2009), *Complexity: A Guided Tour*, Oxford University Press.（复杂, 湖南科技出版社, 2011）

Moses, Joel (2008), Macsyma: A Personal History, Invited Presentation in Milestones in Computer Algebra, May 2008, Tobago, http://esd.mit.edu/Faculty_Pages/moses/Macsyma.pdf.

Moto-Oka, T. (1982), (ed.) *Fifth Generation Computer Systems*, North Holland, 1st reprint 1982, 2nd reprint 1983.

Nerode, Anil and Richard A. Shore (1997), *Logic for Applications,* 2nd ed., Springer.

Neuroscience-Inspired Artificial Intelligence, *Neuron*.

Newborn, M. (1975), *Computer Chess*, Academic Press.

Newborn, M. (2002), *Deep Blue: An Artificial Intelligence Milestone*, Springer.

Newell, Alan (1981), Intellectual Issues in the History of AI, in *The Study of Information*, ed. Machlup and Mansfield.

Newell, Alan (1991), Oral History Interview with Alan Newell, Charles Babbage Institute, http://conservancy.umn.edu/handle/11299/107544.

Nilsson, Nils J. (1989), Oral history interview with Nils J. Nilsson. Charles Babbage Institute, http://hdl.handle.net/11299/107545.

Nilsson, Nils J. (2010), *The Quest for Artificial Intelligence: A History of Ideas and Achievements*, Cambridge University Press.

Overback, R. and E. Lusk, Wos (1997), *Automated Deduction at ANL: The Ethos*, in Veroff (ed.).

Penrose, Roger (1989), *The Emperor's New Mind*, Oxford University Press.（皇帝的新脑, 湖南科技出版社, 2007, 这个译本不推荐）

Petzold, C. (2008), *The Annotated Turing*, John Wiley & Sons.（图灵的秘密：他的生平、思想及论文解读, 人民邮电出版社, 2012）

Pinker, Steve (1997), *How the Mind Works*.

Plaisted, David A. (2015), History and Prospects for First-Order Automated Deduction, CADE-25, *Lecture Notes in Computer Science*, Vol. 9195, pp. 3-28.

Poundstone, William (1992), *Prisoner's Dilemma: John von Neumann, Game Theory, and the Puzzle of the Bomb*.（囚徒的困境, 中信出版社, 2016）

Prawitz, Dag, Håkan Prawitz and Neri Voghera (1960), A mechanical proof procedure and its realization in an electronic computer. *Journal of the ACM*, 7(2):102-128.

Putnam, Hilary (1981), *Reason, Truth, and History*.（理性, 真理与历史, 上海译文出版社, 1997）

Quine, van Orman Willard (1955), A proof procedure for quantification theory. *Journal of Symbolic Logic* 20, 141–149.

Reddy, Raj (1991), Oral history interview with Raj Reddy, Charles Babbage Institute, http://hdl.handle.net/11299/107605.

Reeke, George N., Jr. and Gerald M. Edelman (1988), Real Brain and Artificial Intelligence, in Graubard.

Riazanov, Alexandre (2002), The design and implementation of VAMPIRE, PhD Thesis, University of Manchester.

Robinson, Alan J. (1965), A machine-oriented logic based on the resolution principle. *Journal of ACM* 12, pp. 23–41.

Robinson, Alan J. (1992), Logic and Logic Programming, *CACM*, Vol. 35 March.

Robinson, Alan J. and Andrei Voronkov (2001), *Handbook of Automated Reasoning*, 2 vols.

Rodney A. Brooks (2002), *Flesh and Machines: How Robots Will Change Us*, Vintage Books.

Rosenblatt, Frank (1958), The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386.

Rudolph, Sebastian (2011), Foundations of Description Logics, In *Reasoning Web: Semantic Technologies for the Web of Data, 7th International Summer School, Lecture Notes in Computer Science,* Vol. 6848, pp. 76–136. Springer.

Ruelle, D. (2007), *The Mathematician's Brain: A Personal Tour Through the Essentials of Mathematics and Some of the Great Minds Behind Them*, Princeton University Press.

Russell, Stuart and Peter Norvig (2012), *Artificial Intelligence: A Modern Approach*, 3rd ed.

Samuel, A. (1960), Programming computers to play games. *Advances in Computers,* 1.

Samuel, A. (1967), Some studies in machine learning using the game of checkers.

Schaeffer, Jonathan, et al (2007), Checkers Is Solved, *Science*, Vol. 317, pp. 1518–1522, September 14.

Sergot, Marek (2002), Bob Kowalski: A Portrait, in Kakas, A. C. and F. Sadri.

Shannon, Claude E. (1950), A Chess Playing Machine, *Scientific American*, Feb. 1950.

Shannon, Claude E. (1950), Programming a Computer for Playing Chess, *Philosophical Magazine*, Ser.7, Vol. 41, No. 314, March 1950.

Shannon, Claude E. (1982), Oral History, http://ethw.org/Oral-History: Claude_E._Shannon.

Shannon, Clause and John McCarthy (1956), *Automata Studies*, Princeton University Press.

Shapiro, Ehud (1987), (ed.) *Concurrent Prolog: Collected Papers*, MIT Press.

Shasha, D. and Lazere, C. (2010), *Natural Computing: DNA, Quantum Bits, and the Future of Smart Machines*, W. W. Norton & Company.（自然计算：DNA, 量子比特和智能机器的未来, 人民邮电出版社, 2014）

Shasha, Dennis E. and Cathy Lazere (1998), *Out of their Minds: The Lives and Discoveries of 15 Great Computer Scientists,* Copernicus.

Shaw, D. E. (1981). NON-VON: A Parallel Machine Architecture for Knowledge Based Systems.

Siekmann, J. and G. Wrightson (1983) (eds.), *Automated Reasoning 2 – Classical Papers on Computational Logic 1967–1970*, Vol. 2, pp. 1–24. Springer, Berlin.

Siekmann, J. and G. Wrightson (1983) (eds.), *Automation of Reasoning — Classical Papers on Computational Logic 1957-1966*, Vol. 1. Springer, Berlin.

Silver, David, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. (2016), Mastering the game of go with deep neural networks and tree search. *Nature*, 529 (7587):484–489.

Simon, Herbert (1996), *The Sciences of Artificial*, 3rd Edition, MIT Press.

Simon, Herbert and Toshinori Munakata (1997), AI Lessons, *Communication of ACM*, August, 1997.

Simon, Herbert (1996), *Models of My Life*, MIT Press.

Sipser, M. (2012), *Introduction to the Theory of Computation*, 3rd ed., Cengage Learning.

Smiley, Jane (2010), *The Man Who Invented the Computer: The Biography of John Atanasoff, Digital Pioneer*.（最强大脑：数字时代的前世今生, 新世界出版社, 2015）

Smith, S. W. (2005), Turing is from Mars, Shannon is from Venus: Computer Science and Computer, *IEEE SECURITY & PRIVACY*.

Solomonoff, Ray (1997), The Discovery of Algorithmic Probability, *Journal of Computer and System Sciences*, Vol. 55, pp. 73-88.

Sowa, John (1993), Book Review of Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project, *Artificial Intelligence*, Vol. 61, pp. 95-104.

Sperlich, Woldfgang (2006), *Noam Chomsky*, Realtion Books.（乔姆斯基, 北京大学出版社）

Spiser, Dag and Kirsten Tashevthe, *The Quest to Build Thinking Machine: A History of Computer Chess,* Computer History Museum.

Standage, Tom (2002), *The Turk: The Life and Times of the Famous Eighteenth- Century*

Chess-Playing Machine. Walker and Company, New York City.

Sutcliffe, Geoff and Christian Suttner, The TPTP Problem Library for Automated Theorem Proving, http://www.cs.miami.edu/~tptp/.

Sutton, R. and A. Barto (1998), *Reinforcement Learning: An Introduction*, 1st ed. 1998, 2nd ed. 2017，MIT Press. （这本书第一版和第二版初稿的在线版：http://incompleteideas.net/sutton/book/the-book.html）

Tarski, Alfred (1948), *A Decision Method for Elementary Algebra and Geometry*.

Turing, A. M. (1953). Digital computers applied to games, in *Faster than Thought*, Bowden, B. V. ed. London, Pitman.

Turing, A. M. (1936), On Computable Numbers, with an Application to the Entscheidungs problem, *Proceedings of the London Mathematical Society*. 2. 42. pp. 230–65.

Turing, A. M. (1937), Computability and λ-Definability, *The Journal of Symbolic Logic*. pp. 153–163.

Turing, A. M. (1950), Turing's contributions to the Conference on Information Theory http://www.turing.org.uk/sources/info50turing.html.

Turing, A. M. (1947), Lecture to the London Mathematical Society on 20 February 1947, in *Cooper and Leeuwen*, 2013.

Turing, A. M. (1948), *Intelligent Machinery*, National Physical Laboratory, Teddington, 1948, in Maltzer, B. and D. Michie (ed.) *Machine Intelligence*, Vol. 5, Edinburgh University Press, Edinburgh, 1969.

Tymoczko, T. (1979), The Four-Color Problem and Its Philosophical Significance, *The Journal of Philosophy*, Vol. LXXVI, No. 2.

Valiant, Leslie (2013), *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World*.

Veroff, R. (1997) (ed.), *Automated Reasoning and Its Applications: Essays in Honor of Larry Wos*, MIT Press.

von Neumann, John (1948), The general and logical theory of automata, in *Cerebral Mechanisms in Behavior: The Hixon Symposium,* also in*Collected Works of von Neumann.*

von Neumann, John (2013), *The Computer and the Brain*, 3rd ed., Yale University Press.（计算机与大脑, 商务印书馆）

Wang, Hao (1960), Proving theorems by pattern recognition—I, *CACM*, Vol. 3, pp. 220-234.

Wang, Hao (1960), Toward Mechanical Mathematics, *IBM J. Research and Development*, Vol. 4. pp. 2-22.

Wang, Hao (1960), Proving theorems by pattern recognition—II, *Bell System Technical Journal*, 40, pp. 1-41.

Wang, Hao (1978), Private communication with Wu Wen-tsun, April 10, 1978.

Wang, Hao (1981), *Popular Lectures on Mathematical Logic.*（数理逻辑通俗讲话, 科学出版社, 1981）

Wang, Hao (1984). Computer Theorem Proving and Artificial Intelligence, reprint in *Computation, Logic and Philosophy, A Collection of Essays*, 1990, originally in *Contemporary Mathematics, AMS.*

Wang, Haohan and Bhiksha Raj (2017), On the Origin of Deep Learning, arXiv: 1702.07800v4 [cs.LG] 3 Mar 2017.

Warren, David H. D. (1982), A View of the Fifth Generation and Its Impact, *AI Magazine*, Fall.

Watkins, Christopher (1989), *Learning from Delayed Rewards*, PhD Thesis, Cambridge University.

Weizenbaum, Joseph (1966), ELIZA—A Computer Program for the Study of Natural Language Communication Between Man And Machine, *Communications of the ACM*, Vol. 9, No. 1.

Weizenbaum, Joseph (1976), *Computer Power and Human Reason: From Judgement to Calculation.*

Widrow, Bernard (2013), Oral History of Bernard Widrow, http://archive.computerhistory.org/resources/access/text/2014/01/102746758-05-01-acc.pdf, Computer History Museum, Mountain View, California.

Wiener, Nobert (1964), *I Am a Mathematician*, MIT Press.

Wiener, Nobert (2013), *Cybernetics: Or the Control and Communication in the Animal and the Machine,* 1st ed 1948, 2nd ed. 2013.

Winograd, Terry (1972), *Understanding Natural Language*, Academic Press.

Winograd, Terry (1982), *Language As A Cognitive Process*, Vol. 1, Syntax Addison-Wesley.

Winograd, Terry (1991), Oral history interview with Terry Allen Winograd. Charles Babbage Institute, http://hdl.handle.net/11299/107717.

Winograd, Terry and Fernando Flores (1986), *Understanding Computers and Cognition: A New Foundation for Design*, Ablex Publ Corp.

Winston, Patrick (1990), Oral history interview with Patrick H. Winston, Charles Babbage Institute, http://conservancy.umn.edu/handle/11299/107719.

Wittgenstein, Ludwig (1951), *Philosophical Investigations.*

Wolfram, Stephen (2002), *A New Kind of Science.*

Wolfram, Stephen (2017), *Idea Makers: Personal Perspectives on the Lives and Ideas of Some Notable People,* Wolfram Media.

Wos, L. and Henschen, L. (1983), *Automated theorem proving 1965–1970*. In Siekmann, J., Wrightson, G. (eds.) *Automated Reasoning 2 – Classical Papers on Computational Logic 1967–1970,* Vol. 2, pp. 1–24. Springer, Berlin, 1983.

Wos, L., R. Overbeck and E. Lusk (1991), Subsumption, a Sometimes Undervalued Procedure, in *Lassez and Plotkin*.

Wos, L., R. Overbeck, E. Lusk and J. Boyle (1983), *Automated Reasoning: Introduction and Applications,* McGraw-Hill.

Wos, L. (2013), *The Legacy of a Great Researcher*, in Bonacina eds. 2013.

Wright, R. (1988), *Three Scientists and Their Gods*, Times Books.

Wu, Wen-tsun (2001), *Mathematics Mechanization*, Science Press and Kluwer Academic Publishers.

Wu, Yonghui, et al (2016), Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, *Technical Report.*

Yourgrau, Palle (2005), *A World Without Time: The Forgotten Legacy of Godel and Eistein*, Basic Books.（没有时间的世界, 电子工业出版社）

Yukota, Kazumasa (1994), From Databases to Knowledge-Bases, Proceedings of FGCS 1994, *ICOT*, Tokyo.

Zhang, Nick (1992), Complexity of Neural Network Learning in Real Number Model, *Proceedings of 2nd IEEE Workshop on Physics and Computation*, Dallas, Texas.

冯志伟 (2008)，计算语言学基础，商务印书馆.

尼克 (2014)，哲学评书，浙江大学出版社.

吴文俊 (1984)，几何定理机器证明的基本原理，科学出版社.

吴文俊 (2003)，数学机械化，科学出版社.

吴文俊 (2015)，走自己的路：吴文俊口述自传，湖南教育出版社.

政协齐河县文史资料委员会 (2000)，齐河文史资料第六辑：科学巨匠王浩.

周志华 (2016)，机器学习，清华大学出版社.

人名对照

A

阿比卜　Michael Arbib
阿达玛　Jacques Solomon Hadamard
阿伦森　Scott Aaronson
阿佩尔　Kenneth Appel
阿什比　Ross Ashby
埃德尔曼　Gerald Edelman
埃克特　John Presper Eckert
艾伦　Mary Ellen
爱波斯坦　David Epstein
爱因斯坦　Albert Einstein
安德鲁斯　Peter B. Andrews
安德森　Marc Andreessen
奥克利　Brian Oakley
奥斯丁　John Langshaw Austin
奥威尔　George Orwell

B

阿乌林·布卢姆　Avrim Blum
巴克斯　John Warner Backus
巴罗　Harry Barrow
巴托　Andy Barto
白贝尔　Wolfgang Bibel
柏林纳　Hans Berliner
邦迪　Alan Bundy
贝尔曼　Bellman
本迪克斯　Peter Bendix
本内特　Charles Bennett
比森　Michael Beeson
波尔　Robert S. Boyer
波普尔　Karl Popper
伯恩斯坦　Alex Bernstein
伯克霍夫　George David Birkhoff
伯克斯　Authur Burks
伯林　Isaiah Berlin
伯奈斯　Paul Bernays
博斯特罗姆　Nick Bolstrom
博伊德　Cassidy Boyd
布坎南　Bruce Buchanan
布莱克　Archie Blake
布莱森　Bill Bryson
布莱索　Woody Bledsoe
布利斯博格　Mojżesz Presburger
布林　Sergey Brin
布鲁克斯　Rodney Brooks
布鲁姆　Manual Blum

布什　Vannevar Bush
德布罗金　Nicolaas Govert de Bruijn
雷诺尔・布卢姆　Lenore Blum

C

蔡汀　Gregory Chaitin
查普曼　John Chapman

D

戴客斯特拉　Edsger Wybe Dijkstra
戴森　Freeman Dyson
戴维斯　Martin Davis
丹尼特　Daniel Dennett
道本　Joseph Dauben
道金斯　Richard Dawkins
德布罗金　de Brujin
德施维茨　Nachum Dershowitz
丁斯利　Marion Tinsley
丢东聂 Jean Alexandre Eugène Dieudonné
杜彻斯内　Ricardo Duchesne
多依奇　David Deutsch
斯图亚特・德雷弗斯　Stuart Dreyfus
休伯特・德雷弗斯　Hubert Dreyfus
翟若适　Carl Djerassi

E

爱之地　Donald W. Loveland
厄埃　Gérard Huet
厄内斯特　Les Earnest

F

菲尔德曼　Julian Feldman
菲舍尔　Bobby Fischer
菲亭　Melvin Fitting
费佛曼　Solomon Feferman
费根鲍姆　Edward Albert Feigenbaum
费马　Pierre de Fermat
费曼　Richard Phillips Feynman
费米　Enrico Fermi
费舍　Ronald Fisher
冯诺伊曼　John von Neumann
弗班克　Furbank
弗兰肯斯坦　Frankenstein
弗雷德伯格　R. M. Friedberg
弗雷德金　Edward Fredkin
弗里格　Gottlob Frege
弗洛雷斯　Fernando Flores
伏尼契　Ethel Lilian Voynich
福吉　Charles Forgy

G

甘迪　Robin Gandy
甘金格　Harald Ganzinger
高德纳　Donald Ervin Knuth
哥德尔　Kurt Gödel
格兰特　Herb Gelernter
格雷汉姆　Ron Graham
格林布拉特　Richard Greenblatt
格鲁什科　Robert Glushko
根岑　Gerhard Gentzen
古德　Irving John Good
古德曼　Nelson Goodman
古德瑞克　Oded Goldreich
古德斯丁　Herman Goldstine
古哈　Ramanathan V. Guha

H

埃尔布朗 Jacques Herbrand
哈泊 Grace Hopper
哈尔莫斯 Paul Halmos
哈尔西 William Halsey
哈里斯 Zellig Harris
海尔梅尔 George Heilmeirer
海思 Patrick Hayes
韩丁 William Hinton
寒春 Joan Hinton
赫布 Donald Hebb
赫拉利 Yuval Harari
黑肯 Wolfgang Haken
亨培尔 Carl Hempel
亨亭顿 Samuel Huntington
胡贝尔 David Hubel
胡塞尔 Edmund Husserl
怀特海 Alfred North Whitehead
霍根 John Horgan
霍兰德 John Holland
霍普 Grace Murray Hopper
霍普菲尔德 John Hopfield
霍奇 Wilfrid Hodges
霍维茨 Eric Horvitz
希尔伯特 David Hilbert

I

英曼 Inman

J

吉尔莫 Paul Gilmore
贾德 Steve Judd
贾里尼克 Frederick Jelinek
杰洛姆 Jean-Léon Gérôme

K

卡尔纳普 Paul Rudolf Carnap
卡尔森 Magnus Carlsen
卡普 Richard Karp
卡普尔 Deepak Kapur
卡斯帕罗夫 Garry Kasparov
坎普尔 Murray Campbell
康斯特波 Robert L. Constable
康韦 Conway
考夫曼 Matt Kaufman
柯尔莫格罗夫 Kolmogorov
柯克 John Cocke
柯里 Haskell Curry
科德 Edgar Codd
科恩 Paul Joseph Cohen
科夫 Rich Korf
科林斯 George E. Collins
科莫饶尔 Alain Colmerauer
科瓦尔斯基 Robert Kowalski
克拉克 Jim Clark
克拉克 Edmund M. Clarke
克劳福 Harry Klopf
克劳斯 Lawrence M. Krauss
克里克 Francis Crick
克里尼 Stephen Kleene
克里普克 Saul Aaron Kripke
克里切利 Simon Critchley
克门尼 John Kemeny
克柔克 Steve Crocker
肯德尔 Eric Kandel
肯佩伦 Wolfgang von Kempelen
寇恩 Paul Cohen

寇克 John Cocke
寇普兰 Jack Copeland
寇扎 John Koza
库布里克 Stanley Kubrick
库恩 Thomas Sammual Kuhn
库克 Stephen Arthur Cook
库兹韦尔 Ray Kurzweil
蒯因 Willard Van Orman Quine

L

阿兰・罗宾逊 John Alan Robinson
伯纳斯-李 Tim Berners-Lee
拉希德 Rick Rashid
莱夫谢茨 Solomon Lefschetz
莱瑟 Victor Lesser
朗道 Lev Davidovich Landau
朗道尔 Rolf Landauer
雷蒂 Raj Reddy
雷纳特 Douglas Lenat
李德伯格 Joshua Lederberger
里克 George Reeke
利克莱德 Joseph Licklider
列维 David Levy
列文 Leonid Levin
卢梭 Roussel
鲁梅尔哈特 David Rumelhart
罗宾斯 Herbert Robbins
罗切斯特 Nathaniel Rochester
罗森布拉特 Frank Rosenblatt
伯特兰・罗素 Bertrand Russell
斯图尔特・罗素 Stuart Russell
乔治・罗宾逊 George Robinson
亚伯拉罕・罗宾逊 Abraham Robinson

M

马尔 David Marr
马库恩 William McCune
马纳 Zohar Manna
马泽尔 Johann Nepomuk Maelzel
麦卡洛克 Warren McCulloch
麦卡锡 John McCarthy
麦克达克 Pamela McCorduck
麦克道高尔 Duncan MacDougall
麦克利兰德 James McLelland
曼斯菲尔德 Mike Mansfield
芒福德 David Mumford
毛彻里 John Mauchly
梅尔泽 Bernard Meltzer
蒙克 Ray Monk
米德 Carver Mead
米勒 George Miller
米奇 Donald Michie
明斯基 Marvin Lee Minsky
摩尔 J Strother Moore
摩尔 Trenchard More
摩尔康 Christopher Morcom
摩根斯顿 Oskar Morgenstern
莫里斯 Ian Morris
莫瑞 Arnold Murray

N

冯诺伊曼 John von Neumann
纳什 John Nash
内尔逊 C Greg Nelson
牛伯恩 Monty Newborn
纽厄尔 Allen Newell
纽曼 Judith Newman

诺伯格 Norberg
诺曼 Don Norman
诺维格 Peter Norvig

O

欧赫 Franz Josef Och
欧文白克 Overback

P

庞加莱 Jules Henri Poincaré
庞提 Merleau-Ponty
佩里 Matthew Perry
佩珀特 Seymour Papert
佩奇 Larry Page
彭罗斯 Roger Penrose
皮茨 Walter Pitts
皮亚诺 Giuseppe Peano
平克 Steve Pinker
坡斯特 Emil Post
珀里思 Alan Perlis
普拉格维茨 Dag Pragwitz
普雷斯泰 David Plaisted
普林茨 Dietrich Prinz
普特南 Hilary Putnam

Q

乔姆斯基 Avram Noam Chomsky
丘奇 Alonzo Church

R

瑞迪 Raj Reddy

S

萨顿 Richard Sutton
萨瑟兰 Ivan Edward Sutherland
萨斯曼 Sussman
萨特克里夫 Geoff Sutfliffe
塞尔 John Searle
塞弗里奇 Oliver Selfridge
塞缪尔 Arthur Samuel
塞耶斯 Dorothy Sayers
沙布 Michael Shub
舍佛 Jonathan Schaeffer
施伦伯杰 Schlumberger
史蒂克尔 Mark E. Stickel
史密斯 S. W. Smith
司考特 Dana Scott
司马贺 Herbert Simon
斯金纳 Burrhus Frederic Skinner
斯梅尔 Steve Smale
斯帕斯基 Boris Spassky
斯坦因 Aurel Stein
斯特拉切 Christopher Strachey
所罗门诺夫 Solomonoff
索瓦 John Sowa
肖特莱福 Edward Shortliffe

T

马蹄・塔南鲍姆 Marty Tennenbaum
塞吉诺斯基 Terry Sejnowski
塔尔斯基 Alfred Tarski
塔克 Albert Tucker
汤普森 Ken Thompson
特蕾西娅 Maria Theresia
铁木钦科 Thomas Tymoczko

本书全面讲述人工智能的发展史，几乎覆盖人工智能学科的所有领域，包括人工智能的起源、自动定理证明、专家系统、神经网络、自然语言处理、遗传算法、深度学习、强化学习、超级智能、哲学问题和未来趋势等，以宏阔的视野和生动的语言，对人工智能进行了全面回顾和深度点评。

本书作者和书中诸多人物或为师友或相熟相知，除了详实的考证还有有趣的轶事。本书既适合专业人士了解人工智能鲜为人知的历史，也适合对人工智能感兴趣的大众读者作为入门的向导。

图灵社区：iTuring.cn
反馈/投稿/推荐邮箱：contact@turingbook.com
读者热线：(010) 51095186-600
分类建议 计算机/人工智能
人民邮电出版社网址：www.ptpress.com.cn

ISBN 978-7-115-47160-4
9 787115 471604 >
ISBN 978-7-115-47160-4
定价：49.00元

看完了

如果您对本书内容有疑问，可发邮件至 contact@turingbook.com，会有编辑或作译者协助答疑。也可访问图灵社区，参与本书讨论。

如果是有关电子书的建议或问题，请联系专用客服邮箱：ebook@turingbook.com。

在这可以找到我们：

微博 @图灵教育 : 好书、活动每日播报
微博 @图灵社区 : 电子书和好文章的消息
微博 @图灵新知 : 图灵教育的科普小组
微信 图灵访谈 : ituring_interview，讲述码农精彩人生
微信 图灵教育 : turingbooks